CGP

GCSE OCR
Computer Science

For the Grade 9-1 Exams

New!

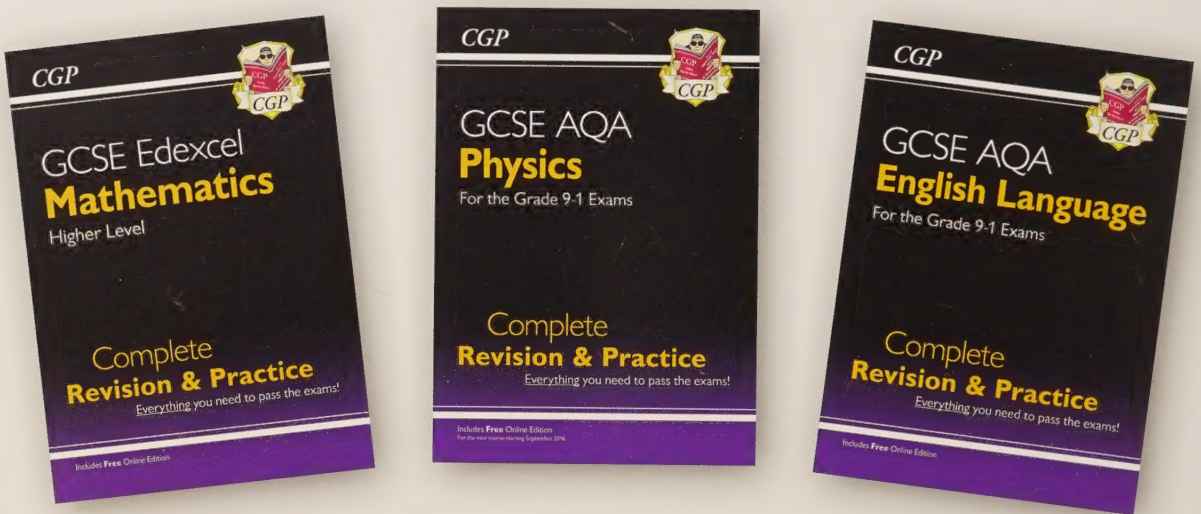
Complete
Revision & Practice

Everything you need to pass the exams!

Includes **Free** Online Edition

CGP

The best that money can buy...

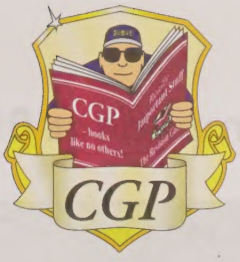


...from the country's leading producer of revision material.

- "CGP sell more revision books than anyone else."
- "Over 90% of all schools use CGP revision products as an integral part of their courses."
- "CGP books are by far the most concise, relevant, clear and informative revision books you will find anywhere."

"Before we got CGP books we didn't realise learning could be quite so much fun."

Dorothy McGuire (left) and Miriam Messenger
Business Studies Students



GCSE OCR
Computer Science

No doubt about it — OCR's Grade 9-1 GCSE Computer Science exams are seriously tough to crack. But this CGP book is the perfect user guide...

It's full of detailed notes, clear diagrams and useful examples, not to mention plenty of practice questions to test your performance.

There's also a full mock exam, with answers and a complete mark scheme, so there'll be no compatibility errors when you finally face the real thing.

How to access your free Online Edition

This book includes a free Online Edition to read on your PC, Mac or tablet.
You'll just need to go to **cgpbooks.co.uk/extras** and enter this code:

4308 2678 9601 1718

By the way, this code only works for one person. If somebody else has used this book before you, they might have already claimed the Online Edition.

Complete
Revision & Practice

Everything you need to pass the exams!

Contents

Component 01 — Computer Systems

Section One — Components of a Computer System

Section Two — Networks

Section Three — Issues

Contents

Component 02 — Computational Thinking, Algorithms and Programming

Section Four — Algorithms

Section Five — Programming

Section Six — Design, Testing and IDEs

Section Seven — Data Representation

Published by CGP

Editors: Sammy El-Bahrawy, Rob Harrison, Shaun Harrogate, Jack Perry.

Contributors: Colin Harber-Stuart, Shaun Whorton.

With thanks to Simon Little for the proofreading.

With thanks to Ana Pungartnik for the copyright research.

ISBN: 978 1 78294 860 5

Printed by Elanders Ltd, Newcastle upon Tyne.
Clipart from Corel®

Based on the classic CGP style created by Richard Parsons.

Computer Systems

As it's the first page I'll start simple. Computer Science is all about computers. What, you already knew that?

A Computer is a Machine that Processes Data

1) The purpose of a computer is to take data, process it, then output it. Computers were created to help process data and complete tasks more efficiently than humans.
2) A computer system consists of hardware and software that work together to process data/complete tasks.

- Hardware is the physical stuff that makes up your computer system, like the CPU, motherboard, monitor and printer.
- Software is the programs or applications that a computer system runs e.g. an operating system, a word processor or video game.

External pieces of hardware like the keyboard, mouse and printer are called peripherals.

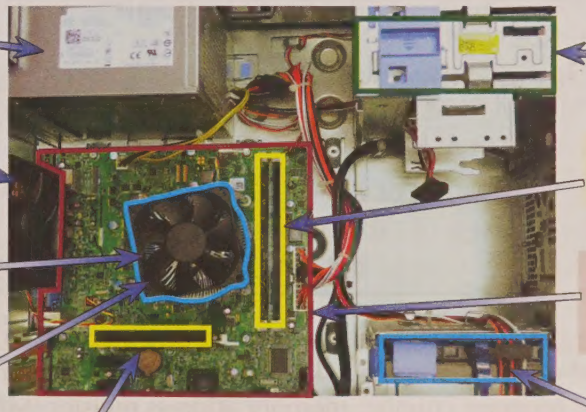
3) There are many types of computer system. These range from small devices like calculators and watches, up to large supercomputers used by banks or for scientific applications. Computers may be general purpose (designed to perform many tasks, e.g. PCs and tablets) or dedicated systems (designed for one particular function, e.g. controlling traffic lights or an aeroplane).

Embedded Systems are Computers inside a Larger System

1) Embedded systems are computers built into other devices, like dishwashers, microwaves and TVs. They are usually dedicated systems.
2) Embedded systems are often used as control systems — they monitor and control machinery in order to achieve a desired result. E.g. In a dishwasher the embedded system could control the water pumps and water release mechanisms, manage the various dishwasher cycles and control the thermostat to keep the water at an appropriate temperature.
3) As they're dedicated to a single task, embedded systems are usually easier to design, cheaper to produce, and more efficient at doing their task than a general purpose computer.

Computers contain Components which Work Together

This section is all about the main hardware components of a computer.
As a warm-up, let's take a look inside a typical desktop PC.



Power supply — supplies power to motherboard, optical and hard drives, and other hardware.

Case cooling fan — extracts hot air from the computer case.

CPU heat sink and cooling fan — keeps the CPU at a steady temperature (CPUs generate a lot of heat).

CPU (hidden under the heat sink) — the most important component. Does all the processing (see p.2-3).

Optical drive — for read/writing of optical discs (see p.9).

RAM sticks (computer memory) slot in here (see p.6-7).

Motherboard — The main circuit board in the computer, where the hardware is connected.

Hard Disk Drive — Internal secondary storage (see p.8).

The graphics card slots in here (see p.7).

If you know your computer, you need not fear defeat...

There's a lot to take in on this first page. You should make sure you're comfortable with the components on this page before going any further, as they'll crop up a lot throughout this section.

The CPU

The CPU is very important — it's the main component of a computer, so here are two whole pages about it.

The CPU is the Central Processing Unit

1) The CPU is the brain of the computer system.
2) It processes all of the data and instructions that make the system work.
3) The processing power of a CPU depends on different characteristics, like its clock speed, number of cores and cache size — there's lots about this on p.7.
4) The CPU architecture describes the main components of the CPU, how they interact with each other, and with other parts of the computer system. Von Neumann and Harvard are the two main types of architecture. You will need to know about Von Neumann — see next page.

CPUs contain 1000s of gold pins — some of these transmit data, others supply power to the CPU.

The CPU has Three Main Parts

The Control Unit (CU)

- The control unit is in overall control of the CPU. Its main job is to execute program instructions by following the fetch-decode-execute cycle (see next page).
- It controls the flow of data inside the CPU (to registers, ALU, cache — see below.) and outside the CPU (to main memory and input/output devices).

The Arithmetic Logic Unit (ALU)

- The ALU basically does all the calculations.
- It completes simple addition and subtraction, compares the size of numbers and can do multiplications and divisions using repeated addition and subtraction.
- It performs logic operations such as AND, OR and NOT (see p.98) and binary shifts (see p.105) — remember, computers process binary data.
- It contains the accumulator register — see next page.

The Cache

- The cache is very fast memory in the CPU. It's slower than the registers (see below), but faster than RAM (see p.6).
- It stores regularly used data so that the CPU can access it quickly the next time it's needed. When the CPU requests data, it checks the cache first to see if the data is there. If not, it will fetch it from RAM.
- Caches have a very low capacity and are expensive compared to RAM and secondary storage.
- There are different levels of cache memory — L1, L2 and L3. L1 is quickest but has the lowest capacity. L2 is slower than L1 but can hold more. L3 is slower than L2 but can hold more.

The CPU contains various registers which temporarily hold tiny bits of data needed by the CPU. They are super-quick to read/write to, much quicker than any other form of memory. You need to know about the program counter, memory address register (MAR), memory data register (MDR) and the accumulator (see next page).

REVISION TASK

That's a lot to remember, for something so small...

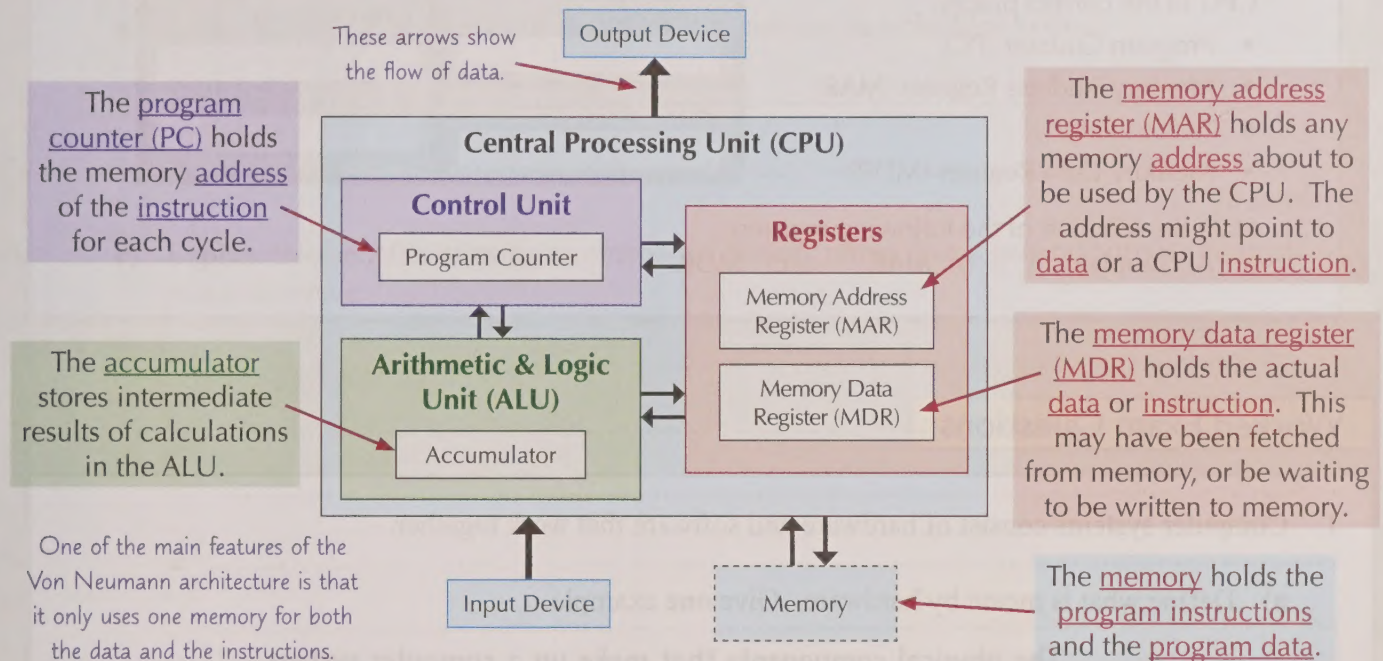
It's important that you know all about the CU, ALU and cache. Try learning everything you can about each one, then cover up the page and write down as many notes as you can remember.

The CPU

Now let's look at the Von Neumann architecture and what the registers do in a bit more detail.
Von Neumann came up with his design in 1945 and it still describes how most computers work today.

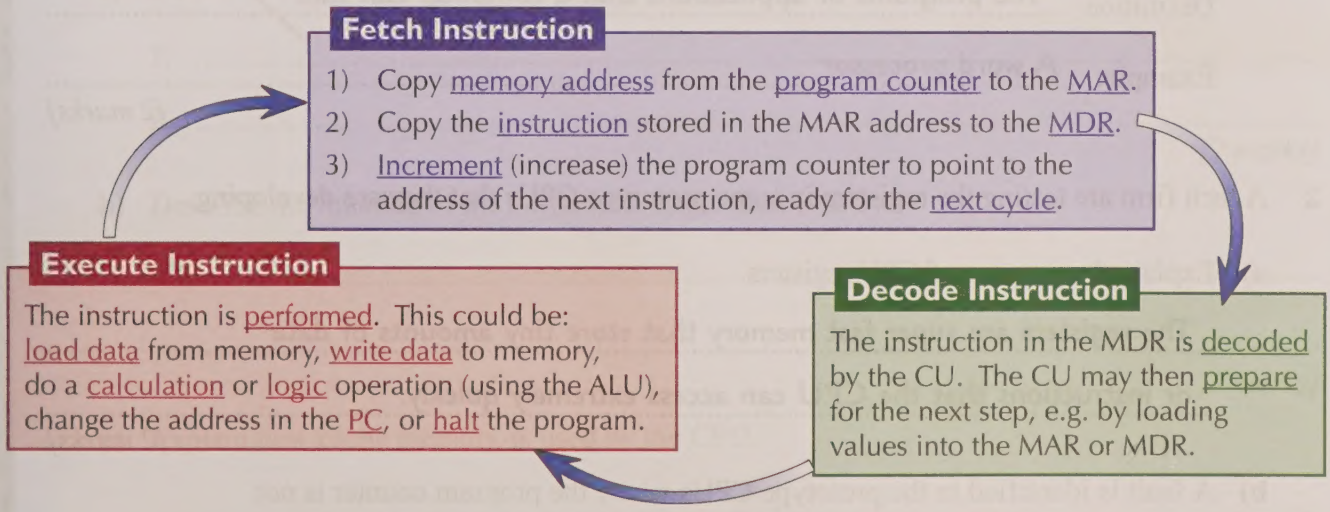
Von Neumann's Design **Revolutionised** Computing

The Von Neumann architecture describes a system where the CPU runs programs stored in memory.
Programs consist of instructions and data which are stored in memory addresses.



CPUs follow the **Fetch-Decode-Execute Cycle**

Essentially, all a CPU does is carry out instructions, one after another, billions of times a second.
The Fetch-Decode-Execute cycle describes how it does it.



REVISION TIP

Learn this with the Revise-Assess-Review cycle...

To remember what each register does, look at its name to see if it stores an address or data. If you're confused about the difference between the PC and MAR, remember: the program counter just starts off the cycle by pointing to the instruction. The MAR is far busier — all addresses (data or instruction) being used must go into the MAR, meaning its value might change several times each cycle.

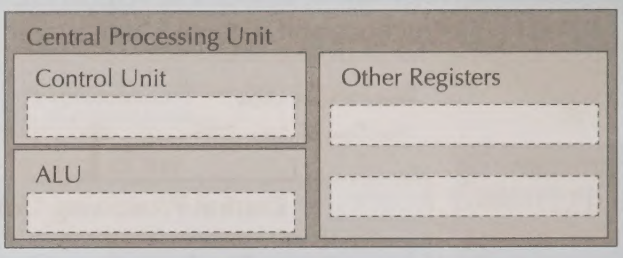
Warm-Up and Worked Exam Questions

There's a lot to learn on those first few pages — once you're happy with them, have a go at these questions.

Warm-Up Questions

1) Name five hardware components of a typical desktop computer.

2) Write the names of the parts of the CPU in the correct places:
   - Program Counter (PC)
   - Memory Address Register (MAR)
   - Accumulator
   - Memory Data Register (MDR)



3) State the function of the following registers:
   a) Accumulator b) MAR c) MDR

Worked Exam Questions

1 Computer systems consist of hardware and software that work together.

a) Define what is meant by hardware. Give **one** example.

Definition: The physical components that make up a computer system.

Example: A mouse

[2 marks]

There are loads of possible examples for each of these — far too many to list.

b) Define what is meant by software. Give **one** example.

Definition: The programs or applications that a computer can run.

Example: A word processor

[2 marks]

2 A tech firm are testing the registers in some prototype CPUs that they are developing.

a) Explain the purpose of CPU registers.

The registers are super fast memory that store tiny amounts of data or instructions that the CPU can access extremely quickly.

[2 marks]

b) A fault is identified in the prototype CPUs where the program counter is not incrementing with each cycle. Explain what will happen in the CPU in this case.

The memory address of the next instruction will always be the same, so the CPU will carry out the same instruction repeatedly.

[2 marks]

Exam Questions

3 A microwave contains an embedded system which controls its cooking modes.

a) What is an embedded system?

.....

[1 mark]

b) Give **two** other examples of devices that may contain an embedded system.

1. .....

2. .....

[2 marks]

c) Explain **two** benefits of using an embedded system, rather than a general purpose computer, in a microwave.

1. .....

.....

.....

2. .....

.....

.....

[4 marks]

4 The control unit, arithmetic logic unit and cache memory are all parts of the CPU.

a) State **two** functions of the Control Unit.

1. .....

2. .....

[2 marks]

b) Describe the function of the Arithmetic Logic Unit (ALU).

.....

.....

[2 marks]

c) Explain how cache memory is used by the CPU.

.....

.....

.....

[3 marks]

Memory

As you'll have gathered from earlier, memory is a pretty fundamental part of a computer. It contains all the instructions that the CPU follows. Without memory, a computer wouldn't know what to do with itself.

RAM is High Speed, Volatile memory

1) RAM (or Random Access Memory) is used as the main memory in a computer. It can be read and written to. RAM is volatile.

- Volatile memory is temporary memory. It requires power to retain its data.
- Non-volatile is permanent memory — it keeps its contents even when it has no power.

2) The main memory is where all data, files and programs are stored while they're being used.
3) When a computer boots up, the operating system is copied from secondary storage to RAM.
4) When software applications, documents and files are opened, they are copied from secondary storage to RAM. They stay in RAM until the files or applications are closed.
5) RAM is slower than the CPU cache, but much faster than secondary storage.

Secondary storage is covered on p.8-9.

Virtual Memory is Secondary Storage used as extra RAM

1) Computers have a limited amount of RAM. As applications are opened, RAM fills with data.
2) When RAM is full, the computer needs somewhere else to put application data. It moves data that hasn't been used recently to a location on secondary storage (p.8) known as virtual memory.
3) Virtual memory may be needed if there are too many applications open at once, or if a particularly memory-intensive application is being used (or both).
4) If the CPU needs to read data stored in virtual memory, it must move the data back to RAM. This is slow as data transfer rates are much slower on secondary storage than RAM.
5) Using virtual memory can make a computer slow to respond when switching between applications (while data for one application in virtual memory is swapped with the other) or when using a memory-intensive application (due to data constantly moving between virtual memory and RAM just to keep the program running).

ROM tells the CPU how to Boot Up

1) ROM ('Read Only Memory') is non-volatile memory. As it says on the tin, it can only be read, not written to.
2) ROM comes on a small, factory-made chip built into the motherboard.
3) It contains all the instructions a computer needs to properly boot up. These instructions are called the BIOS (Basic Input Output System).
4) As soon as the computer is powered on, the CPU reads the instructions from ROM. This tells the CPU to perform self checks and set up the computer, e.g. test the memory is working OK, see what hardware is present and copy the operating system into RAM.
5) Although the CPU can only read ROM, it is possible to update ('flash') the BIOS on a ROM chip.

The BIOS is a type of firmware — hardware-specific software built in to a device. Embedded systems (p.1) are controlled by firmware.

ROM chips often use flash memory. This is a very common type of non-volatile memory that stores data in electrical circuits by trapping electrons. It's used in SD cards, USB sticks and solid state drives (SSDs). There's loads about flash devices on p.8.

Get all this information stored on your brain's non-volatile memory...

RAM is where the computer puts everything it's working on. Don't confuse memory with secondary storage — if a computer has a 2 TB (see p.100) hard drive, that doesn't mean it has 2 TB of memory.

CPU and System Performance

All sorts of things affect the speed of a computer system, but the biggest factors are usually to do with the hardware. Choice of CPU, RAM and GPU (see below) can all have big effects on performance.

CPU Performance depends on Clock Speed, Cores and Cache

Clock speed

- This is the number of instructions a single processor core can carry out per second (Hz). For most desktop computers, this will be somewhere around 3.5 GHz (3.5 billion instructions per second).
- The higher the clock speed, the greater the number of instructions that can be carried out per second.
- Some CPUs can be overclocked to make them run at a higher clock speed than the factory-set rate. But it's risky if not done properly — it can make CPUs overheat, causing crashes or permanent damage to the system. High performance cooling systems (e.g. water cooling) are usually needed.

Number of Cores

- Each core in a CPU can process data independently of the rest.
- The more cores a CPU has, the more instructions it can carry out at once, so the faster it can process a batch of data.
- Most PCs and smartphones have 4 or more cores these days.

It's not quite as simple as 'doubling the number of cores doubles performance'. Software needs to be designed to use multicore processing. And not all processing tasks can be split evenly between cores — some steps will depend on others, meaning one core may end up waiting for another core to catch up.

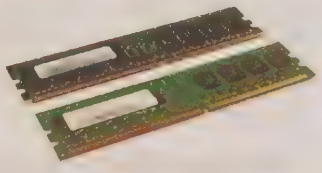
Cache Size

- The cache (p.2) is data storage inside the CPU that's much faster than RAM.
- A larger CPU cache gives the CPU faster access to more data it needs to process.

Generally speaking, CPUs with higher clock speeds, more cores or larger caches will have better performance, but will also be more expensive.

More RAM can mean a Faster or Smoother System

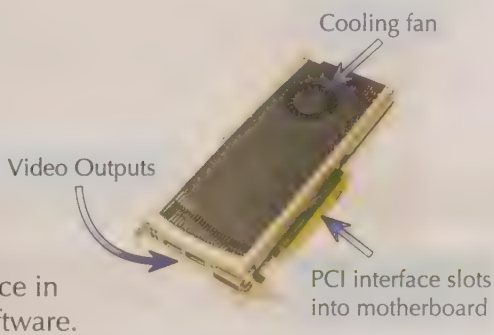
1) If a computer has too little RAM it may run slowly due to the use of virtual memory (see previous page).
2) The more RAM, the more applications or more memory-intensive applications it can smoothly run, making it faster overall.
3) It's easy to upgrade RAM on a PC or laptop — it's just a matter of replacing the RAM sticks with higher capacity (or higher speed) ones.
4) If the computer already has plenty of RAM to run everything the user wants, increasing RAM may make no difference to performance.



RAM comes on sticks which plug into slots on the motherboard.

GPUs help CPUs process Images

1) GPUs (graphics processing units) are specialised circuits for handling graphics and image processing. They relieve the processing load on the CPU, freeing it to do other things.
2) Computers have basic GPUs integrated onto the motherboard or the CPU. For better graphics performance, a dedicated GPU (graphics card) is often used.
3) Using high-end graphics cards can greatly improve performance in graphics-intensive applications, e.g. PC gaming and design software.



High-end hardware tends to be very expensive...

There are other factors that affect CPU performance, but you don't need to worry about them at GCSE. Using SSDs rather than traditional hard drives is another way to speed up a computer — more info on p.8.

Secondary Storage

When you think of memory, you might think of USB sticks, CDs, hard drives, etc. — these are all types of 'secondary storage'. All that RAM and ROM stuff from p.6 was actually what we call 'primary storage'.

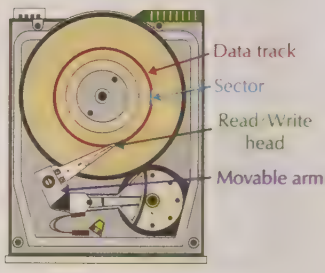
There are Two Main Tiers of Storage

1) Primary storage refers to the memory areas that the CPU can access very quickly, like CPU registers, cache, ROM and RAM. Primary storage has the fastest read/write speeds and is mostly volatile (p.6).
2) Secondary storage is non-volatile — it's where all data (operating systems, applications and user files) are stored when not in use. It includes magnetic hard disk drives, solid state drives, CDs and SD cards. Read/write speeds are much slower compared to primary storage.

There's also tertiary storage, which is used for long term data storage (it's mainly used for archives and back-ups of massive amounts of data).

Magnetic Hard Disks are High-Capacity, Reliable Storage

1) Hard disk drives (HDDs) are the traditional internal storage in PCs and laptops.
2) A hard disk drive is made up of a stack of magnetised metal disks spinning at a rate of between 5400 and 15000 rpm (revolutions per minute).
3) Data is stored magnetically in small areas called sectors within circular tracks. Read/write heads on a moving arm are used to access sectors on the disks.
4) Portable HDDs are popular for backing up and transporting large amounts of data.
5) Despite their moving parts, HDDs are generally very long lasting and reliable, although they could be damaged by large impacts like being dropped.



Solid State Drives are Fast and Reliable Secondary Storage

1) Solid State Drives (SSDs) are storage devices with no moving parts. Most of them use a type of flash memory (see p.6). SSDs are used for the same purpose as HDDs — for internal storage.
2) SSDs have significantly faster read/write times than HDDs. Using a SSD rather than traditional HDD can give much quicker times for booting up and opening programs and files.
3) Hybrid drives exist which use solid state storage for the OS and programs, and a hard disk for data.
4) Like HDDs, portable SSDs can be used to back up and transport data.

Other types of flash storage

USB pen drives and memory cards (e.g. SD cards) are also flash-based, solid-state storage.

They're much slower than SSDs and have a much shorter read/write life.

They're used to expand the storage capacity of small devices like cameras, smartphones and tablets (which are too small for SSDs or HDDs). Their capacity is very high relative to their tiny size.

There are Advantages to using HDDs and SSDs

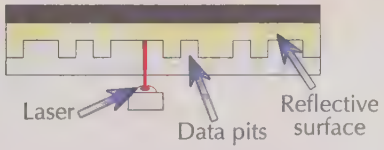
Advantages of HDDs	Advantages of SSDs
• HDDs are cheaper.	• SSDs are faster.
• Both are high capacity, but HDDs are higher.	• SSDs don't need defragmenting (see p.15).
• HDDs have a longer read/write life than SSDs — SSDs can only be written a certain number of times before they begin to deteriorate.	• SSDs are more shock-proof than HDDs.
	• HDDs make some noise, SSDs are silent.

Secondary Storage

Optical Discs are Cheap and Robust Secondary Storage

1) Optical discs are things like CDs, DVDs and Blu-Ray™ discs.
2) CDs can hold around 700 MB of data, DVDs can hold around 4.7 GB and Blu-Rays can hold around 25 GB.
3) Optical discs come in three forms:

   – read-only (e.g. CD-ROM / DVD-ROM / BD-ROM)
   – write-once (e.g. CD-R / DVD-R / BD-R)
   – rewritable (e.g. CD-RW / DVD-RW / BD-RW)



Data is stored as microscopic indentations on the shiny surface of the disc. Data is read by shining a laser beam on the surface and detecting changes in the position of the reflected beam.

4) Nowadays, their use is declining:
   - As Internet speeds have increased, streaming and download services like Netflix®, iTunes® and Steam® have removed the need for optical discs.
   - Modern devices like phones and tablets don't have optical drives.
   - DVD-Rs and DVD-RWs used to be popular for backing up data, but they can't compete with flash storage devices due to their low capacity per disc, very slow read/write speeds and poor reliability of RW discs.
5) They do have some advantages — they're very cheap (per GB), portable, and won't be damaged by water or shocks (although they are easily scratched).

Magnetic Tapes are used for Archiving

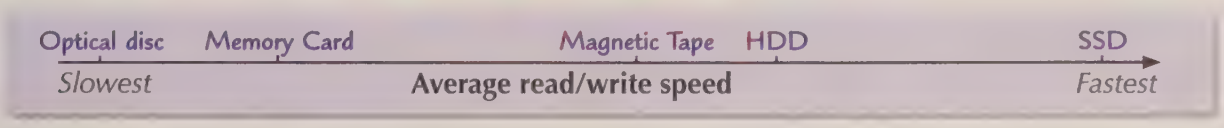
1) Magnetic tape has much greater storage capacity than HDDs. It also has an extremely low cost per GB.
2) Magnetic tapes are often used by large organisations in archive libraries to store huge amounts of data.
3) It comes in plastic cassettes (containing reels of tape). Cassettes require a special tape-drive for read/writing.

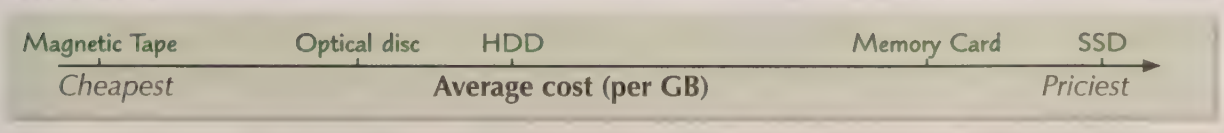
Magnetic tapes are most suitable for businesses who do large, frequent back-ups — see p.15.

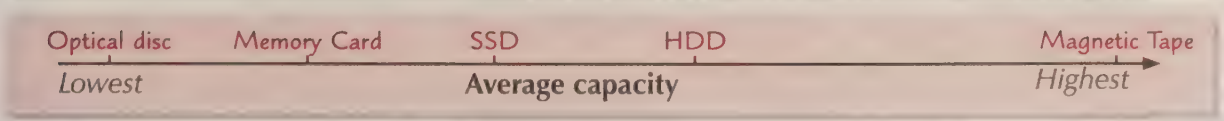
4) Tape is read/written sequentially, meaning it is read/written from the beginning to the end, or until it is stopped by the computer. This means tape is very slow when finding specific data stored on it, but has a fast read/write speed once it is in the correct place to begin reading/writing.

A quick Summary...

Here's a summary of the relative speeds, costs and capacities of all these different types of storage.







REVISION TIP

Be sure to learn the pros and cons of all these types of storage...

Be careful with your terminology. Storage media refers to the actual thing that holds the data, e.g. an optical disc. Storage devices read/write data to media, e.g. optical drives, or HDDs.

Warm-Up and Worked Exam Questions

Once you've got all the different memory and storage terms learnt, it's time to have a go at some questions. If you're confident with the warm-up questions, then test yourself against the exam questions.

Warm-Up Questions

1) What is the difference between RAM and ROM?
2) Explain what it means for a single-core processor to have "a clock speed of 3 GHz".
3) Name a type of secondary storage that:
   a) usually comes on a reel in a cassette
   b) stores data as little pits on its surface
   c) stores data on a stack of magnetic disks
4) Give one advantage of storing data on a USB pen drive over an optical disk.

Worked Exam Questions

1 Nigella runs a piece of software to analyse the performance of her computer. It recommends that she should install more RAM in her computer.

a) State the purpose of RAM in a computer system.

RAM holds any data that is currently in use.

[1 mark]

b) Give **two** reasons why Nigella may need to install more RAM in her computer.

1. Her computer may be running slowly.
2. She may want to run more programs at once.

[2 marks]

2 Shaun is on a skiing trip. Each night, he copies skiing videos to his laptop's secondary storage.

a) Give **three** characteristics to consider when choosing a suitable type of secondary storage for a computer system.

You could also mention how quickly they transfer data, or how portable they are.

1. Capacity
2. Cost
3. Durability — Durability is how much physical damage it can take without breaking.

[3 marks]

b) Shaun uses a helmet-mounted action camera while skiing, which records onto a flash memory card. Give **two** reasons why this is a suitable storage type for an action camera.

1. Flash storage is resistant to impacts, so is unlikely to be damaged when the action camera is in use.
2. Flash storage can be very compact and lightweight.

[2 marks]

Exam Questions

3 Diana has bought a new laptop. The laptop contains 3 GB RAM and 128 GB secondary storage.

a) Explain why secondary storage is needed in addition to RAM.

.....

.....

.....

[3 marks]

b) Diana wants to back up the data on her laptop twice a week.
Give **two** advantages and **two** disadvantages of storing her backup data on optical discs.

Advantages: 1. .....

2. .....

Disadvantages: 1. .....

2. .....

[4 marks]

4 Jackson is considering upgrading his PC. Will offers to sell his old CPU to Jackson.
Will's CPU is the same type as Jackson's CPU but has a different specification.

Jackson's CPU
8 cores
6 MB cache
1.6 GHz clock speed

Will's CPU
4 cores
3 MB cache
2.8 GHz clock speed

a) Explain why using a CPU with a large cache capacity may increase CPU performance.

.....

.....

[2 marks]

b) Do you think Jackson should buy Will's CPU? Give reasons to justify your decision.

.....

.....

.....

.....

There's no wrong answer, as long as your answer is properly justified. *[4 marks]*

c) Jackson increases the RAM in his PC from 4 GB to 8 GB, but is disappointed to find no noticeable increase in his computer's performance. Explain why this may be the case.

.....

.....

[2 marks]

Systems Software — The OS

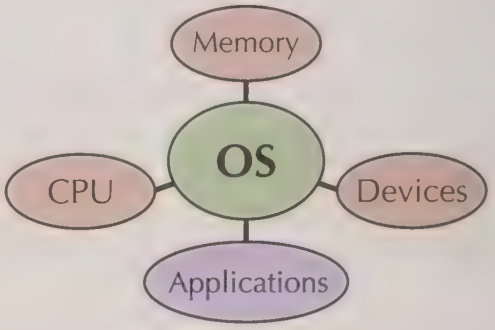
Systems software is software designed to run and maintain a computer system. By far the most important one is the operating system (OS). There's also utility software (see p.15), but that's very much the runner up.

Operating Systems manage Hardware and run Software

An Operating System (OS) is a complex piece of software found on most computer systems.

Main Functions of an OS

- Communicate with internal and external hardware via the device drivers (see below).
- Provide a user interface, allowing a user to interact with the computer (see p.13).
- Provide a platform for different applications to run (see p.13).
- Allow the computer to multi-task by controlling memory resources and the CPU (see p.13).
- Deal with file management and disk management (see p.14).
- Manage the security of the system, e.g. through user accounts (see p.14).

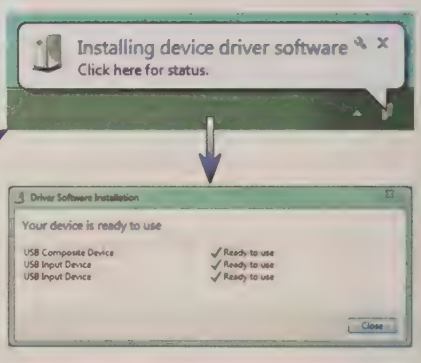


The words 'application' and 'program' can be used interchangeably to describe computer software.

Device Drivers let the OS and Hardware Talk to Each Other

Operating systems use device driver software to communicate with internal hardware or peripherals connected to the computer system:

- Every piece of hardware connected to the computer system requires a device driver. Drivers essentially act as a 'translator' for the signals between OS and hardware.
- When a computer is booted up, the OS will choose the correct device drivers for the hardware it detects. If new hardware is connected to the computer, the system will install the new, matching driver.
- Device manufacturers may release updates to device drivers in order to fix bugs, add features or improve the performance of their hardware. Updates may be installed automatically by the OS or manually by the user.



Some OSs will automatically find and install the drivers when you plug in a new device.

A driver lets a computer speak to a mouse? If you say so...

The OS is the boss of the computer system. It gives you a way to interact with your computer, controls hardware via drivers, and allows the computer to run applications and multi-task. There's loads more to learn about the OS, so make sure you've got your head around everything on this page before moving on.

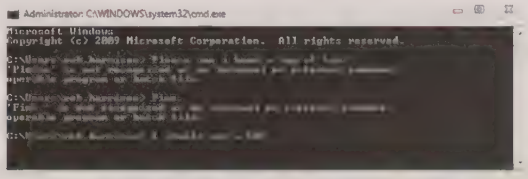
Systems Software — The OS

One of the most recognisable functions of an OS is its user interface. It's one of the first things you imagine when you think about a particular system. But there's much more to an operating system than how it looks...

Operating Systems provide a **User Interface**

1) A User Interface allows the user to interact with a computer system.
2) Graphical User Interfaces (GUIs) are the most common type — they're designed to be easy for everyday users by making them visual, interactive and intuitive.
3) GUI systems are optimised for specific input methods. In the past, GUIs have been WIMP-based (using windows, icons, menus and pointers). Android™ and iOS® were created for touchscreen devices, using finger gestures like pinching and swiping in place of a mouse.
4) A command-line interface is text based.
The user enters specific commands to complete tasks.
Command-line interfaces are less resource-heavy than GUIs.
5) Command-line interfaces aren't suitable for everyday users. But for advanced users, they can be far more efficient and powerful than a GUI. They can be used to automate processes using scripts (simple programs).

You can swipe between screens or tap an icon to open it on Android™.



You type instructions into a command-line interface.

The OS allows **Multi-Tasking** by managing **Resources**

1) Operating Systems provide a platform to run applications (by configuring hardware so they can use it, and giving access to the CPU and memory).
2) Operating Systems that can run multiple applications at the same time are called multi-tasking OSs.
3) The OS helps the CPU carry out multi-tasking by efficiently managing memory and CPU processing time:

- When an application is opened, the OS moves the necessary parts of the application to memory, followed by additional parts when they are required. The OS will decide if applications or features have been used recently — if not, they may be removed from memory.
- To run multiple applications, the OS needs to make sure that the applications don't overwrite or interfere with each other. A memory manager allocates certain applications certain memory addresses, to make sure their processes are placed into separate locations.
- Only one application is processed by the CPU at a time, so the other processes must wait. The OS divides CPU time between open applications and may prioritise certain processes in order for instructions to be executed in the most efficient order.
- When required, the OS organises the movement of data to and from virtual memory.

It may seem like lots of programs are processed at the same time, but in reality the CPU switches between each one extremely quickly.

Make sure you understand how multi-tasking works...

Take a look at a few different types of OS (e.g. desktop computer, smartphone, game console). What differences can you see between the GUIs? How have they been adapted to suit the device?

Systems Software — The OS

It's also the job of the OS to make sure that all your files are where they're meant to be. So when you go looking for carKeys.txt or tvRemote.exe, they'll be right where you left them.

The OS handles **File** and **Disk Management**

1) Computers store data as files. Images, music, videos and spreadsheets are all just collections of data. File extensions (for example .jpg, .mp3, .mpeg) tell the computer which software should be used to open the file.

2) The OS is responsible for file management — the organisation of data into a usable hierarchical structure. It also deals with the movement, editing and deletion of data.

3) The hard disk is also managed by the OS. It splits the physical disk into storage sectors, decides which sectors to write data to, and keeps track of free space on the disk. Ideally, the data for a single file would be placed in adjacent sectors, but this isn't always possible (see p.15).

4) The OS may also include utility software to help it manage files and disks. File compression software can reduce the size of individual files and encryption software is used to secure the contents of files. Defragmentation software can help to organise and maintain the hard disk by collecting all the free space together.

Utility software can be used for loads of different things. For example, File Explorer allows users to navigate and edit the file structure or access their files. See p.15 for more about utilities.

Operating Systems deal with **User Accounts**

1) Operating Systems can be single-user or multi-user.

- Single-user operating systems allow only one user to use the computer at once. Most common OSs, such as Windows 10® and OS X®, are single-user operating systems, even if the computer has multiple user accounts, or is connected to a network (see p.20).
- Multi-user OSs (e.g. UNIX server) allow several users to use the computer at the same time. They're often used on mainframes (huge supercomputers) and give many users simultaneous access. For example, ATMs allow thousands of people access to a large bank's mainframe at the same time.

2) The OS is also responsible for user account control. User accounts allow different users to be granted access to specific data or resources on a computer system.

3) On most desktop operating systems each user has access to their own personal data and desktop, but cannot access other users' personal data.

4) Operating systems may have anti-theft measures to prevent other users from accessing locked devices or accounts to steal information. User accounts may be password, or pin protected. Some devices also require a user to draw a specific pattern on the screen, or have fingerprint or retina scanners.

Two people at one computer doesn't count as a multi-user OS...

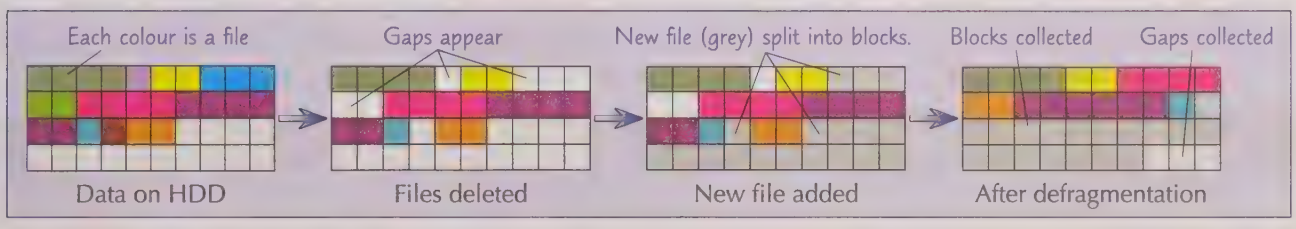
Make sure you know the difference between multi-user and single-user OSs. A single user OS may have multiple user accounts, but only a single person can use the computer at any one time.

Systems Software — Utilities

Utility system software helps to maintain or configure a computer. Many useful utilities are installed with the operating system, but extra utility software can be installed to perform additional tasks.

Defragmentation Utilities put broken up files back together

1) Files are stored on a hard disk in available spaces. Ideally, entire files would be stored together.
2) However, as files are moved, deleted and change size, lots of small gaps begin to appear on the disk. When writing files to the disk, the OS splits files into smaller blocks to fill up the gaps.
3) Over time, the disk becomes more and more fragmented. This makes reading and writing files slower as the read/write head has to move back and forth across the disk.
4) Defragmentation software reorganises data on the hard drive to put fragmented files back together. It also moves files to collect all the free space together. This helps to prevent further fragmentation.



5) As SSDs use flash storage with no moving parts, fragmentation doesn't cause them any problems — they can access data just as quickly however it's arranged. In fact, as SSDs have a limited number of read/writes, defragmenting them can actually shorten their lifespan.

Backup Utilities help to... Backup data

1) A backup is a copy of a computer system's files and settings stored externally. This means data can be recovered in the event of data loss. Data loss can happen for many reasons: fire, theft, flood, malware, hardware failure, or even just accidentally deleting a file.
2) A backup utility is software with facilities such as scheduling of regular backups, creating rescue disks, disk images, and options for full or incremental backups:

A full backup is where a copy is taken of every file on the system. They often use a lot of storage space. A full backup can take a long time to create, but is faster to restore from.

To restore from full backups, only the latest back-up is needed.

Incremental backups are where only the files created or edited since the last backup are copied. They use less storage space and are much quicker to create. But, a full system restore is slow — the last full backup must be restored, followed by every incremental backup since that point.

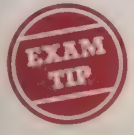
Compression Software

Compression software reduces the size of files so they take up less disk space. It's used loads on the Internet to make files quicker to download. Standard file formats include .zip and .rar. Compressed files need to be extracted before they can be used.

Encryption Software

Encryption software scrambles (encrypts) data to stop third-parties from accessing it. Encrypted data can be decrypted using a special 'key'.

There's more about compression on p.113 and encryption on p.33.



There are many more examples of utility software...

Remember, utilities are bits of software that help maintain your system. You need to know the four above, but it's worth knowing some others for the exam, e.g. disk/registry cleaners, system restore, file managers, anti-virus / anti-spyware / firewalls, automatic updating, system diagnosis tools.

Open Source and Proprietary Software

Generally speaking, open source software is free and proprietary software is paid for.
But the proper definitions are to do with whether the licence lets you change and share the software.

Open Source software is given away with its Source Code

1) Open source software is software where the source code is made freely available. Users may legally modify the source code to create their own spin-off software, which can be shared under the same licence and terms as the original software.
2) Well-known examples include Apache HTTP server (runs web servers), GIMP (image editing), Mozilla® Firefox® (web browser), and VLC media player.
3) Linux is a hugely successful open source OS released in 1991. Hundreds of Linux-based OSs have been developed and shared over the years. The most popular include UBUNTU, Debian and Android™.
4) Popular open-source software is always supported by a strong online community (forums of users sharing ideas and solving problems). Users actively help to improve software — anyone can play with the source code and suggest bug fixes and improvements to the original developers.

Source code is the actual programming code behind the software. It shows exactly how the software was made.

Advantages of Open Source Software

- It is (usually) free.
- Made for the greater good, not profit — it benefits everyone, encourages collaboration, sharing of ideas.
- Software can be adapted by users to fit their needs.
- Wide pool of collaborators can be more creative and innovative than the programmers of one company.
- Popular software is very reliable and secure — any problems are quickly solved by the community.

Disadvantages of Open Source Software

- Small projects may not get regular updates...
- ...and so could be buggy
- ...or have unpatched security holes.
- There may be limited user documentation.
- No warranties if something goes wrong.
- No customer support (although community forums will often make up for this).
- Companies using open-source code to make custom software may not want competitors to see their source code, but they have no choice.

Proprietary Software is Closed Source Software

1) Proprietary software is software, usually paid for, where only the compiled code is released. The source code is usually a closely-guarded secret. Proprietary software licenses restrict the modification, copying and redistribution of the software.
2) Businesses often use proprietary software instead of open source as it tends to have better customer support options. Companies producing proprietary software include Microsoft® (Office®, Windows®, Outlook®, etc.) and Adobe® (Photoshop®, Illustrator®, etc.).

Compiled code is the final file (e.g. .exe file) that runs — it doesn't tell you how the program was made.

Advantages of Proprietary Software

- Comes with warranties, documentation, and customer support.
- Should be well-tested and reliable as the company's reputation depends on this. Fixes and updates will come regularly (open source will vary more).
- Usually cheaper for companies than developing their own custom-built software.

Disadvantages of Proprietary Software

- Can be expensive.
- Software may not exactly fit a user's needs, and they can't do anything about it.
- Software companies may not maintain older software after warranties expire — they'll want people to buy their latest product.

Learn the definitions of open source and proprietary software...
... as well as the pros and cons of each. Just talking about cost is unlikely to get you full marks in the exam.

Warm-Up and Worked Exam Questions

You've made it to the end of Section One, and I think that calls for a bit of a celebration. But not before having a go at these questions — better get them out of the way while it's all fresh in your mind...

Warm-Up Questions

1) Give one way in which an Operating System can help to secure a user's data.
2) What is the purpose of a 'compression utility'?
3) Put the following software into the table on the right:
   - Adobe® Photoshop®
   - Linux
   - Windows® Firewall
   - VLC Media Player
   - macOS®

	Open Source	Proprietary
OS		
Utility		
Other		

Worked Exam Questions

1 The operating system on Selina's computer has an optional command line interface.

a) Identify **two** benefits of using a command line interface instead of a GUI.

1. Command line interfaces give greater control than GUIs.

2. They are less resource-heavy than GUIs.

You could also mention scripts as a possible benefit. *[2 marks]*

b) The operating system includes an encryption utility that can be used to encrypt folders and files. Explain **one** reason why Selina may use the encryption utility.

To keep her data private in the event of a third party, like a hacker, gaining access to her files. It may also protect her data from malware.

[2 marks]

2 A marketing company has the same, paid-for, proprietary software on all of its computers. The software provides facilities for word processing, presentations, spreadsheets and databases.

a) Describe what is meant by proprietary software.

Proprietary software is software where only the compiled code is released. Users are not allowed to modify, copy or redistribute the software.

[2 marks]

b) Give **two** advantages to the company of using proprietary software.

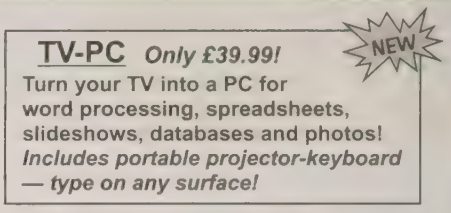
1. The software is likely to include customer support.

2. It should be well-tested and reliable.

[2 marks]

Exam Questions

3 Ioteck has created TV-PCs. TV-PCs plug into any USB-compatible TV, and come packaged with a selection of open source software.



a) Describe what is meant by open source software.

.....

.....

[2 marks]

b) Explain **one** advantage and **one** disadvantage to Ioteck of using open source software on the TV-PCs.

Try to relate your answers specifically to Ioteck's needs.

Advantage: .....

.....

.....

Disadvantage: .....

.....

.....

[4 marks]

4 An accounting firm plans to introduce a new scheme for regularly backing up its data.

a) Define what is meant by the following types of backup.

Full backup: .....

.....

Incremental backup: .....

.....

[2 marks]

b) Describe a possible backup scheme for the firm that includes:

- full backups
- data compression
- incremental backups
- security measures

[4 marks]

Think about when they should take their backups, and what they should do with them afterwards.

5 Josephine's computer has a multi-tasking operating system. Explain how the operating system manages memory and CPU time to allow the computer to multi-task.

[6 marks]

Revision Questions for Section One

Sometimes the first section is the easy one — but not this time. There's lots to learn here, so:

- Try these questions and tick off each one when you get it right.
- When you've done all the questions for a topic and are completely happy with it, tick off the topic.

Computer Systems and the CPU (p.1-3) ☐

1) What is a computer? ☐
2) Define hardware and software. ☐
3) What is a) an embedded system? b) a control system? ☐
4) Explain the role of the control unit in the CPU. ☐
5) Write down the names of three registers in the CPU. ☐
6) What does ALU stand for and what does it do? ☐
7) What is cache memory and what is it used for? ☐
8) Sketch a Von Neumann computer. ☐
9) Describe what happens at each stage of the CPU fetch-decode-execute cycle. ☐

Memory and Computer Performance (p.6-7) ☐

10) What is the difference between volatile and non-volatile memory? ☐
11) What does RAM stand for? Describe how RAM is used in a computer system. ☐
12) Could changing the amount of RAM affect the performance of the computer? Give reasons for your answer. ☐
13) Explain when and how virtual memory is used. ☐
14) Explain why ROM is required by a computer system. ☐
15) Name three characteristics of a processor that may affect its performance. ☐
16) State three components that could be upgraded to speed up a computer system. ☐

Secondary Storage (p.8-9) ☐

17) Define primary and secondary storage. Give an example of each. ☐
18) List four uses of flash memory. ☐
19) List the advantages and disadvantages of HDDs and SSDs. ☐
20) Why might someone choose magnetic tape as a form of storage? ☐
21) What are the pros and cons of optical discs? ☐
22) Draw a diagram to summarise cost, speed and capacity for different types of secondary storage. ☐

Types of Software (p.12-16) ☐

23) List six functions of an operating system. ☐
24) Explain how device drivers are used in a computer system. ☐
25) Briefly describe a GUI and a Command-line interface. ☐
26) Describe how the OS manages resources to allow multi-tasking. ☐
27) What is a multi-user OS? Give an example of where one might be used. ☐
28) List four types of utility software. ☐
29) Explain how defragmentation software works. ☐
30) What are incremental backups and how would they be used? ☐
31) List the advantages and disadvantages for open-source and proprietary software. ☐

Networks — LANs and WANs

When you connect a device to another one, you're creating a network — networks allow devices to share information and resources. Here we'll look at the two types of network you'll need to know for your exam.

A LAN is a Local Area Network

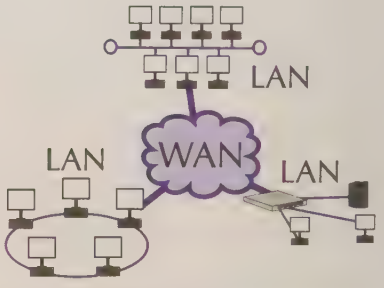
1) A LAN covers a small geographical area located on a single site.
2) All the hardware for a LAN is owned by the organisation that uses it.
3) LANs are either wired (e.g. with Ethernet cables) or wireless (using Wi-Fi®).
4) You'll often find LANs in businesses, schools and universities.
5) Lots of homes have a LAN to connect various devices, such as PCs, tablets, smart TVs and printers.

Why Use A LAN?

- Sharing files is easier — network users can access the same files, work collaboratively on them (at the same time) and copy files between machines.
- You can share the same hardware (like printers) on a LAN.
- The Internet connection can be shared between every device connected to the LAN.
- You can install and update software on all computers at once, rather than one-by-one.
- You can communicate with LAN users cheaply and easily, e.g. with instant messaging.
- User accounts can be stored centrally, so users can log in from any device on the network.

A WAN is a network that Connects LANs

1) WAN stands for Wide Area Network.
2) A WAN connects LANs that are in different geographical locations. For example, a business with offices in three different countries would need a WAN for all their devices to connect together.
3) Unlike a LAN, organisations hire infrastructure (e.g. communication lines) from telecommunications companies, who own and manage the WAN. This is because a WAN is much more expensive to set up than a LAN.



4) WANs may be connected using fibre or copper telephone lines, satellite links or radio links.
5) The Internet is actually one big WAN (see p.30).

Many Factors can affect the Performance of Networks

1) Bandwidth is the amount of data that can be transferred in a given time, e.g. 500 Mbps. The greater the bandwidth, the better the network can perform.
2) Available bandwidth is shared between users of a network — too many users or heavy use (e.g. streaming video) may cause congestion and slow the network. You can limit the bandwidth available to individual users to address this.

Mbps stands for megabits per second, a measure of bandwidth.

3) Wired connections are generally faster and more reliable than wireless. Fibre optic cables can give much better performance than copper cables (see p.21). Wireless performance depends on signal quality so is affected by the range of the device, the amount of interference from other devices and physical obstructions like thick walls in buildings (see p.22).
4) Choice of hardware other than cables (see p.21) and network topology (see p.24) also have a big effect.

Don't LANguish at the bottom of the class — learn this page...

Make sure you're absolutely clear about the differences between LANs and WANs before moving on. Remember, companies use their own cables for LANs but for WANs they almost always hire lines.

Networks — Hardware

Connecting devices doesn't magically happen. To create a network, you need certain pieces of hardware...

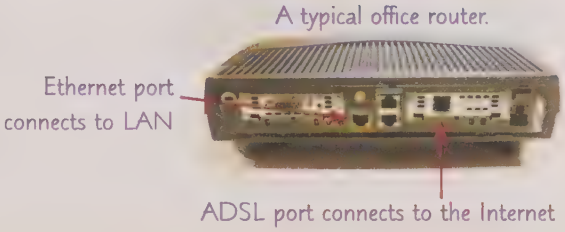
NICs, Switches and Routers

1) A Network Interface Controller (NIC) is an internal piece of hardware that allows a device to connect to a network. These used to be on separate cards, but nowadays they're built into the motherboard. NICs exist for both wired and wireless connections.

2) Switches connect devices on a LAN. Switches receive data (in units called frames) from one device and transmit this data to the device on the network with the correct MAC address (see p.27).

3) Routers are responsible for transmitting data between networks — they're always connected to at least two different networks.

4) Routers have a crucial role on the Internet, directing data (in units called packets) to their destination — see p.28.



5) Routers are used in homes and offices to connect the LAN to the Internet.

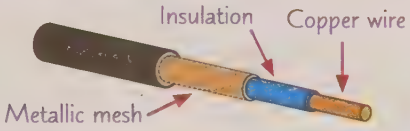
Most home 'routers' are in fact a router, switch and WAP (see p.22) all-in-one.

Cables — CAT 5e/6, Coaxial and Fibre-Optic

1) Ethernet cables are used to connect devices in a LAN. The most common Ethernet cables are CAT 5e and CAT 6. They are 'twisted pair' cables, containing four pairs of copper wires which are twisted together to reduce internal interference.

CAT 6 cable

2) Coaxial cables are made of a single copper wire surrounded by a plastic layer for insulation and a metallic mesh which provides shielding from outside interference.



Coaxial cable

3) Fibre optic cables transmit data as light. They are high performance (and therefore expensive) cables — they don't suffer interference and can transmit over very large distances without loss of signal quality.


Fibre optic cable

Hardware — nothing to do with concrete jackets...

Make sure you understand how switches and routers are different — it might take a while to click.

Wireless Networks

If you don't like the sound of all those cables, don't worry — you can throw them all away and use wireless networking instead. You'll still have to learn about both of them for the exam though, I'm afraid.

Wireless uses Radio Waves to transmit data

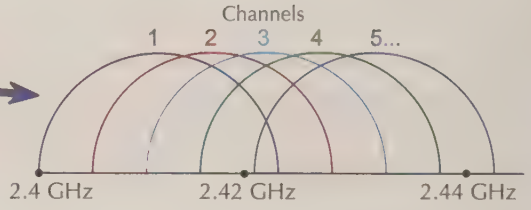
1) Like mobile phones and TVs, wireless networks use radio waves to transmit data.
2) To set up a wireless network, you need a Wireless Access Point (WAP) device. The WAP is basically a switch that allows devices to connect wirelessly.
3) Don't confuse WAPs with hotspots — locations where you can connect to a WAP.

Dongles come in all shapes and sizes (although most of them are roughly the same shape and size).

4) To connect, devices need wireless capability. Many modern devices have the necessary hardware built in, but devices that don't can often still connect to a wireless network using a dongle.
   - USB dongles can be plugged into computers to allow them to connect wirelessly to the internet.
   - HDMI dongles can use wireless networks to stream high-quality video to a TV.

Wi-Fi® is the Standard for Wireless Networks

1) Wi-Fi® uses two radio frequency bands — 2.4 GHz and 5 GHz.
2) The bands are split into numbered channels that each cover a small frequency range. The channels in the 2.4 GHz band overlap.
3) Wi-Fi® performance is affected by interference between networks using adjacent channels. To avoid problems, only certain channels that are spaced apart tend to be used.
4) The frequency band that a network uses can also affect the Wi-Fi® performance:



It's important that data is encrypted (see p.33) on Wi-Fi® networks. The security protocols for this are WPA™ (Wi-Fi® Protected Access) and WPA2™.

Advantages of 2.4 GHz	Advantages of 5 GHz
• Has a greater range, so can serve devices across a wider area. • Better at getting through solid objects/walls.	• Much faster when communicating over a short distance. • There are more non-overlapping channels, so there's less chance of interference from other devices.

REVISION TIP

Remember, Wi-Fi® is a separate thing from the Internet...

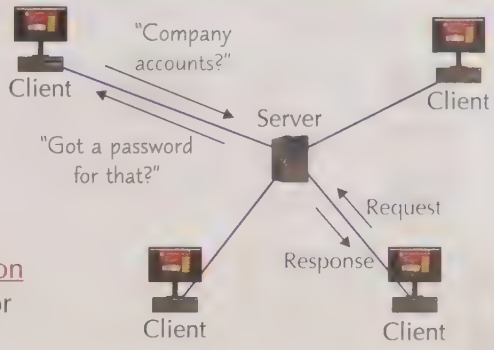
Wi-Fi® allows a device to wirelessly connect to a network with a WAP — that network would then need to be connected to a router in order for the device to be able to access the Internet.

Client-server and Peer-to-Peer Networks

Don't think we're finished with networks yet — we've barely started. If you want to set up a network, you need to decide what form the network is going to take — and whether you're going to need a server or not.

Client-server networks are made up of a Server and Clients

1) A client-server network is managed by a server. The devices connected to the server are clients.
2) Files and software are usually stored centrally on the server rather than on individual client devices.
3) Clients send requests to the server, e.g. asking for data. The server processes the request and responds. This is the client-server relationship.
4) The server stores user profiles, passwords and access information — it may request a password before fulfilling certain requests or deny requests to users without the right access level.



5) Most uses of the Internet work on a client-server relationship. E.g. websites are hosted on web servers. Web browsers are client programs which send requests to web servers. Web servers fulfil requests (e.g. by sending web pages) for thousands (or hundreds of thousands) of clients.

Pros

- Easier to keep track of files as they are stored centrally.
- Easier to perform back-ups.
- Easier to install and update software.
- Easier to manage network security (e.g. anti-malware software and user access levels).
- Servers are very reliable and are always on.

Cons

- Expensive to set up and needs IT specialists to maintain the network and server.
- Server dependence — if the server goes down all clients lose access to their work.
- The server may become overloaded if too many clients are accessing it at once.

Peer-to-Peer networks don't use servers

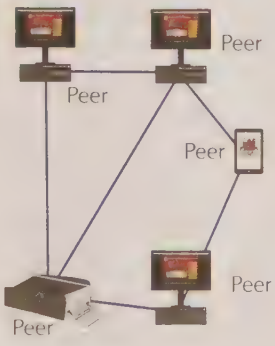
1) In Peer-to-Peer (P2P) networks all devices are equal, connecting directly to each other without a server.
2) You store files on individual devices and share them with others.
3) You may use a P2P network at home to share files between devices, or connect devices to a printer.

Pros

- Easy to maintain — you don't need any expertise or expensive hardware.
- No dependence on server — if one device fails the whole network isn't lost.

Cons

- No centralised management — devices need their updates and security installed individually. Backups are also more complicated.
- Copying files between devices creates duplicate files — it's easy to lose track of what's stored where and which files are up-to-date.
- Peer machines are less reliable and data may be lost if one fails.
- Machines are prone to slow down when other devices access them.



Although most Internet use is client-server based, there are some common P2P applications such as video calling (like Skype™) and file sharing (sadly this is often used for illegal sharing of copyrighted material).

REVISION TASK

Luckily, these networks are exactly what their names suggest...

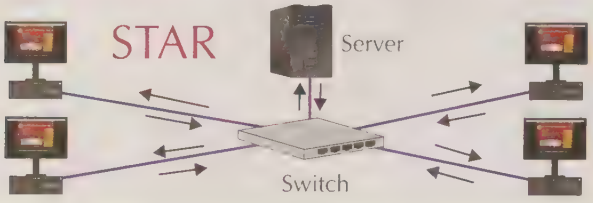
Look into whether the network at home or school is client-server or peer-to-peer. Draw a diagram showing the devices and how they're connected (you can simplify it if there are a lot of devices).

Network Topologies

A topology is essentially the layout of the network. Networks can be arranged in lots of different topologies, but star and mesh are the two important ones you'll need to know for the exam.

In a **Star Topology** all devices are connected to the centre

In a star topology, all the devices are connected to a central switch or server that controls the network.



The central switch allows many devices to access the server simultaneously.

Star networks may be wired or wireless.

Pros

- If a device fails or a cable is disconnected, the rest of the network is unaffected.
- It's simple to add more devices to the network.
- Better performance than other setups — data goes straight to the central device so all devices can transmit data at once (unlike ring network) and there are few data collisions (unlike bus network).

Cons

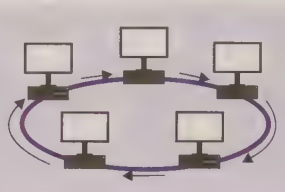
- In wired networks, every device needs a cable to connect to the central switch or server. This can be expensive, e.g. for an office building with 50 terminals.
- If there is a problem with the switch/server, the whole network is affected.

It's worth taking a quick look at some traditional network setups for comparison with star networks:

BUS

In a bus topology, all devices are arranged in a line, connected to a single backbone cable. Devices send data in both directions. This causes data collisions, which slows the network.

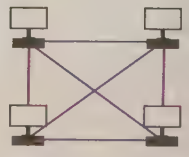
RING



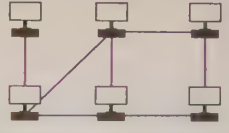
In a ring topology, data moves in one direction around the ring, preventing collisions. But only one device can send data at a time and data passes through many devices before reaching its destination.

In a **Mesh Topology** all devices are connected to each other

1) A mesh topology is a relatively new network layout. It is decentralised — networking devices are either directly or indirectly connected to every other one without the need for one central switch or server. Mesh networks work by sending data along the fastest route from one device to another.
2) The main advantage of a mesh topology is that there is no single point where the network can fail. If the central switch or server of a star network fails then the whole network fails — in a mesh network, if one device fails then the data is sent along a different route to get to its target.
3) The traditional problem with mesh networks has been that they were very expensive — you needed a lot of wire to connect so many devices together. But now more people are using wireless technology, mesh networks are a more practical option.
4) A full mesh topology is where every device is connected to every other device. In a partial mesh topology, not all devices are fully-connected.



Full mesh topology



Partial mesh topology

I tried to set up a star network, but it all ended up a bit of a mesh...

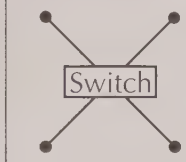
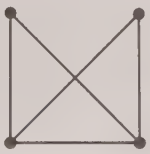
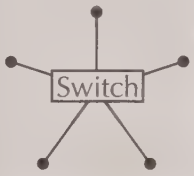
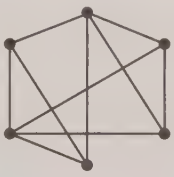
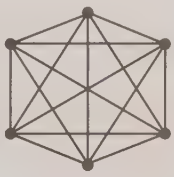
Don't let all this stuff about different types of network confuse you — remember, these diagrams are about the layout of a network, while the client-server/P2P diagrams showed the roles of the devices on the network.

Warm-Up and Worked Exam Questions

Now that you know what networks are made of and what they look like, test yourself against these questions. If anything catches you out, just have another read over it until you're happy.

Warm-Up Questions

1) State one piece of hardware needed in order to set up a LAN.
2) Briefly explain the relationship between client and server in a client-server network.
3) Give one type of software that uses a P2P connection across the Internet.
4) Complete the table showing different network topologies by correctly filling in the white cells:

No. of devices	Star	Partial Mesh	Full Mesh
4			
5			
6			

Worked Exam Question

1 In an office there are six computers, a scanner and a router connected together in a Local Area Network (LAN).

a) Define the term Local Area Network (LAN).

A group of devices connected to share data over a small geographical area.

[1 mark]

b) State **three** advantages of connecting the computers together into a Local Area Network.

1. The business' six computers can share one Internet connection.
2. The computers can share the scanner and any other hardware.
3. It is easier to install and update software.

They can also share files more easily, communicate with instant messaging and store user accounts centrally. *[3 marks]*

Exam Questions

2 Jane connects her laptop to her home LAN using a home router.

a) Jane can connect her laptop to the router using either Ethernet or Wi-Fi®. State how an Ethernet connection is different to a Wi-Fi® connection.

.....

[1 mark]

b) Jane's home router functions as a switch, router and Wireless Access Point (WAP) all in one. State the function of each of these devices.

Switch: .....

Router: .....

WAP: .....

[3 marks]

3 Explain **one** advantage and **one** disadvantage of mesh topologies compared to star topologies.

Advantage: .....

.....

Disadvantage: .....

.....

[4 marks]

4 The staff at a graphic design company work together by sharing files between their computers, which are connected together in a Peer-to-Peer (P2P) network. An IT consultant suggests the company should adopt a Client-Server network setup.

a) Describe what is meant by Peer-to-Peer and Client-Server networks.

Peer-to-Peer: .....

.....

Client-Server: .....

.....

[4 marks]

b) Identify **two** benefits and **two** drawbacks of changing from a Peer-to-Peer (P2P) network to a Client-Server network.

Benefits: 1. .....

2. .....

Drawbacks: 1. .....

2. .....

[4 marks]

Network Protocols

Moving data on a network is like going on a car journey — you need a destination, something to tell you how to get there, and rules to stop you crashing into anyone else on the road. That's where protocols come in.

Networks need **Protocols** to set the rules

1) A protocol is a set of rules for how devices communicate and how data is transmitted across a network.
2) Protocols cover how communication between two devices should start and end, how the data should be organised, and what the devices should do if data goes missing.
See p.29 for more on the different types of protocol.

Communication on the same network uses **MAC Addresses**

1) Every device needs a unique identifier so it can be found on a network.
2) MAC addresses are assigned to all network-enabled devices by the manufacturer. They are unique to the device and cannot be changed.
3) MAC addresses are 48 or 64-bit binary numbers (i.e. a long string of 48 or 64 0s and 1s). To make them easier to use they're converted into hexadecimal.

This binary MAC address is translated into six hexadecimal numbers.

See p.106-107 for more on binary to hex conversion.

4) MAC addresses are mainly used by the Ethernet protocol on LANs. LAN switches read the MAC addresses and use them to direct data to the right device.

Communication between different networks uses **IP Addresses**

1) IP addresses are used when sending data between TCP/IP networks (see p.28-29) e.g. over the Internet.
2) Unlike MAC addresses, IP addresses aren't linked to hardware. They are assigned either manually (static) or automatically (dynamic) before the device can access the network.
3) Static IP addresses are permanent addresses. They're used to connect printers on a LAN, and for hosting websites on the Internet — companies don't want the IP address of their website changing. Static IP addresses on the Internet can be very expensive — businesses pay big money for them.
4) Dynamic IP addresses are assigned when a device logs on to a network, meaning that it may have a different address every time it connects. Internet Service Providers (ISPs) commonly use dynamic IP addresses as they are more cost effective and can be reused.
5) An IP address can either be a 32-bit or 128-bit binary number, depending on the version of IP you're using. The longer 128-bit numbers are translated into eight hexadecimal numbers, while the 32-bit ones are converted into four denary numbers, like in the example below.

This 32-bit binary IP address is translated into four denary numbers.

See p.103 for more on binary to denary conversion.

I hope this useful tIP addresses any confusion you have...

You'll find out lots more about TCP/IP and its uses on page 29. But for now, just remember:
MAC addresses are wired into devices' hardware and are used to communicate within networks.
IP addresses are assigned to devices and are used for communication between networks.

Network Protocols

Quick recap: inside a LAN (Ethernet), data is sent in frames and directed by switches using MAC addresses. Between networks (e.g. over the Internet), data is sent in packets and directed by routers using IP addresses.

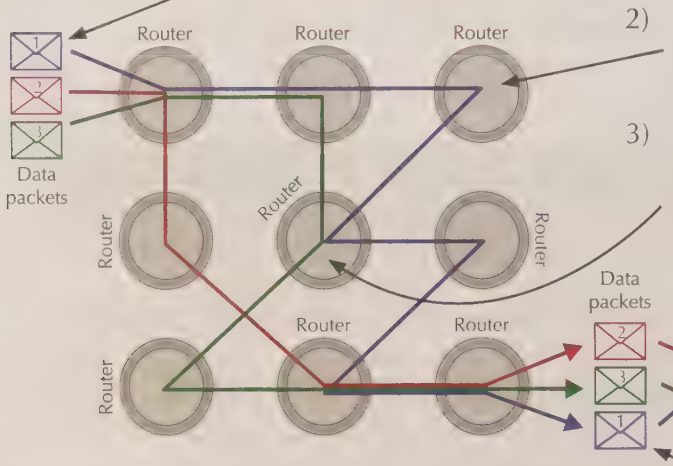
Data is sent between networks in **Packets**

1) Data sent between networks (e.g. on the Internet using TCP/IP) is split into equal-sized packets.
2) Every data packet has a header — this contains the control information. The control information is like the envelope of a letter — it includes the packet's destination address (where it's going), the source address (where it's come from) and the packet number (see below).
3) The data packet's payload is like the letter inside the envelope — it's the thing a person is likely to read, and the whole reason the data packet has been sent in the first place. The payload might be part of an email, document, web page or streamed video.
4) Packets are also likely to include a checksum number — a form of validation used to check that the payload data hasn't been corrupted during transit. The sending and receiving devices both calculate a checksum value by performing a function on the payload data. If the values match then the data has been received correctly.

Packet Switching is used to **Direct** the data

Packet switching is used by routers to direct data packets on the Internet and other IP networks.

1) The sending device splits the data into packets to be sent across the network. Each packet is given a packet number to show the order of the data.



2) Each router reads the packet header and decides which way to send the packet next, according to the IP rules.
3) The way the data is sent changes depending on network traffic so the packets can take different routes. If a router receives too many packets at once, it may prioritise some over others.
4) As the packets take different routes, they can arrive in the wrong order. The receiving device uses the packet numbers to reassemble them in the right order.
5) Sometimes packets go missing in transit, so the receiving device checks periodically that all packets have been received. If it hasn't received them within a certain time, it sends a timeout message back to the sending device.
6) If all the data is received and the checksums match, a receipt confirmation is sent to the sending device.

Packet switching is an efficient use of the network because there are so many possible routes that data can take — packets can reach their receiving device even if there's heavy traffic.

REVISION TIP

They sent me the wrong router — the packets were switched...

You need to be a packet switching master for the exam — keep going over the sequence until it sticks, and sketch the diagram if it'll help you revise. Questions can ask about any part of the process, so make sure you know how packets are sent and received, and what happens on the way.

Network Protocols

This is where we get to the protocols themselves, so it's no surprise that there's a lot of information here.

TCP/IP is the most important protocol

1) TCP/IP is the protocol which dictates how data is sent between networks. It's made up of two protocols.
2) Transmission Control Protocol (TCP) sets the rules for how devices connect on the network. It's in charge of splitting the data into packets and reassembling the packets back into the original data once they reach the receiving device. It's also responsible for checking the data is correctly sent and delivered.
3) Internet Protocol (IP) is responsible for packet switching.
4) Several other protocols build upon TCP/IP to do specific Internet-based tasks:

Protocol	Stands for...	What is it used for?
HTTP	Hyper Text Transfer Protocol	Used by web browsers to access websites and communicate with web servers.
HTTPS	HTTP Secure	A more secure version of HTTP. Encrypts all information sent and received.
FTP	File Transfer Protocol	Used to access, edit and move files between devices on a network, e.g. to access files on a server from a client computer.
POP3	Post Office Protocol — version 3	Used to retrieve emails from a server. The server holds the email until you download it, at which point it is deleted from the server.
IMAP	Internet Message Access Protocol	Used to retrieve emails from a server. The server holds the email until you delete it — you only download a copy. Used by most web-based email clients.
SMTP	Simple Mail Transfer Protocol	Used to send emails. Also used to transfer emails between servers.

Network protocols are divided into Layers

1) A layer is a group of protocols which have similar functions.
2) Layers are self-contained — protocols in each layer do their job without needing to know what's happening in the other layers.
3) Each layer serves the layer above it — it does the hidden work needed for an action on the layer above. E.g. when you send an email (on layer 4), this triggers actions in layer 3, which triggers actions in layer 2, all the way down to layer 1.

Data can only be passed between adjacent layers. E.g. Layer 2 can pass data to Layers 1 and 3 but Layer 1 can only pass data to Layer 2.

Layer Name	Protocols in this layer cover...	Protocol examples
Layer 4 — Application Layer	Turning data into websites and other applications and vice versa.	HTTP, FTP, SMTP
Layer 3 — Transport Layer	Controlling data flow — e.g. splitting data into packets and checking packets are correctly sent and delivered.	TCP
Layer 2 — Network Layer	Making connections between networks, directing data packets and handling traffic. Used by routers.	IP
Layer 1 — Data Link Layer	Passing data over the physical network. Responsible for how bits are sent as electrical signals over cables, wireless and other hardware.	Ethernet

Advantages of Using Layers

- It breaks network communication into manageable pieces. This helps developers concentrate on only one area of the network without having to worry about the others.
- As layers are self-contained, they can be changed without the other layers being affected.
- Having set rules for each layer forces companies to make compatible, universal hardware and software, so different brands will work with each other and always work in basically the same way.

Lots of layers to learn — and plenty of protocols to ponder too...

Make sure you know what the TCP/IP protocol is all about, as well as all the other protocols on this page.

Networks — The Internet

The Internet is so much a part of everyday life, it's easy to forget that it's actually just a really big network.

The Internet is a Worldwide collection of networks

1) The Internet is a network of networks — it's a WAN which connects devices and networks from all over the world. It's based around the protocol TCP/IP.
2) The World Wide Web (WWW) is a collection of websites that are hosted on web servers and accessed through the http protocol.
3) URLs are addresses used to access web servers and resources on them.

URL: https://www.cgpbooks.co.uk/student/books_gcse_cs

- HTTPS is the protocol used (see page 29).
- The domain name of the website. This is linked to an IP address.
- Path to the specific file or page. May also contain a query for the web server.

4) A Domain Name Server (DNS) translates a website's domain name into its IP address. The Internet has a network of Domain Name Servers, meaning you don't need to remember IP addresses to access websites — you can use domain names instead.

The Cloud uses the Internet to store files and applications

1) Hosting is when a business uses its servers to store files of another organisation.
2) The traditional use for this on the Internet is the hosting of websites.
3) A relatively recent use of Internet hosting is for general storage of user files and also providing online software — this is cloud computing, or simply 'the cloud'. It acts like an extension of a traditional client-server network where user files are stored centrally on a network server.

Pros

- Users can access files from any connected device.
- Easy to increase how much storage is available
- No need to buy expensive hardware to store data.
- No need to pay IT staff to manage the hardware.
- Cloud host provides security and back ups for you.
- Cloud software will be updated automatically.

Cons

- Need connection to the Internet to access files.
- Dependent on host for security and back-ups.
- Data in the cloud can be vulnerable to hackers.
- Unclear who has ownership over cloud data.
- Subscription fees for using cloud storage and software may be expensive.

Virtual Networks are software-based networks

1) A virtual network is a network that is entirely software-based. Virtual networks are created by partitioning off some of a physical network's bandwidth to form a separate network.
2) Several virtual networks can exist on the same physical network. These networks all share the same hardware, making virtual networks more efficient than standard physical networks.
3) Each virtual network has its own security, including its own firewall. This means virtual networks can only be accessed by using certain software or login information — other people could be using the same physical network and not have access to the virtual network, or even know that it exists.
4) A Virtual Private Network (VPN) is a type of virtual network that can be used to send data securely over a large network, like a WAN or the Internet. E.g. a VPN can be used to send data securely between two offices on different sites, or to set up a school intranet that all the students can access from home.
5) A Virtual LAN allows you to split a LAN into several separate networks using the same hardware.

REVISION TIP

Learn the difference between the Internet and the WWW...

Don't make the mistake of thinking the WWW and the Internet are the same. The Internet is the network behind the WWW, but it also has many other uses, e.g. email, FTP and instant messaging.

Network Security Threats

Networks are great for lots of reasons, but they can also be very vulnerable to attacks by criminals. Hackers can be very imaginative when it comes to attacking and stealing data stored on networks.

Network **Attacks** come in different forms

1) A passive attack is where someone monitors data travelling on a network and intercepts any sensitive information they find. They use network-monitoring hardware and software such as packet sniffers. Passive attacks are hard to detect as the hacker is quietly listening. The best defence against passive attacks is data encryption (see p.33).

Government agencies sometimes use data interception for cyber-security purposes — this is called Lawful Interception.

2) An active attack is when someone attacks a network with malware (see below) or other planned attacks. They are more easily detected. The main defence against them is a firewall (see p.33).
3) In an insider attack someone within an organisation exploits their network access to steal information.
4) A brute force attack is a type of active attack used to gain information by cracking passwords through trial and error. Brute force attacks use automated software to produce hundreds of likely password combinations, e.g. combining real words with predictable number sequences. Hackers may try lots of passwords against one username or vice versa. Simple measures like locking accounts after a certain number of failed attempts and using strong passwords will reduce the risk of a brute force attack.
5) A denial-of-service attack (DoS) is where a hacker tries to stop users from accessing a part of a network or website. Most DoS attacks involve flooding the network with useless traffic, making the network extremely slow or completely inaccessible.

Malware is software that can harm devices

1) Malware (malicious software) is installed on someone's device without their knowledge or consent.
2) Typical actions of malware include:

- Deleting or modifying files.
- Scareware — e.g. it tells the user their computer is infected with loads of viruses to scare them into following malicious links or paying for problems to be fixed.
- Locking files — ransomware encrypts all the files on a computer. The user receives a message demanding a large sum of money be paid in exchange for a decryption key.
- Spyware — secretly monitors user actions, e.g. key presses, and sends info to the hacker.
- Rootkits alter permissions, giving malware and hackers administrator-level access to devices.
- Opening backdoors — holes in someone's security which can be used for future attacks.

3) Malware can access your device in different ways.

- Viruses attach (by copying themselves) to certain files, e.g. .exe files and autorun scripts. Users spread them by copying infected files and activate them by opening infected files.
- Worms are like viruses but they self-replicate without any user help, meaning they can spread very quickly. They exploit weaknesses in network security.
- Trojans are malware disguised as legitimate software. Unlike viruses and worms, Trojans don't replicate themselves — users install them not realising they have a hidden purpose.

There is more than one way to attack a network...

You don't need to know every single type of malware out there, but learning all of the examples here will give you a great overview of how malware gets in to a system, and the damage it can do once it's there.

Network Security Threats

Often, security threats arise because organisations fail to secure their network properly — they might forget to encrypt their data or use bad code. Other instances are a result of hackers manipulating employees.

People are often the Weak Point in secure systems

1) Social engineering is a way of gaining sensitive information or illegal access to networks by influencing people, usually the employees of large companies.
2) A common form of social engineering takes place over the telephone — someone rings up an employee of a company and pretends to be a network administrator or somebody else within the organisation. The social engineer gains the employee's trust and persuades them to disclose confidential information — this might be personal (e.g. their login details) or sensitive company data.
3) Another type of social engineering is phishing. Phishing is when criminals send emails or texts to people claiming to be from a well-known business, e.g. a bank or online retailer. The emails often contain links to spoof versions of the company's website. They then request that the user update their personal information (e.g. password or bank account details). When the user inputs this data into the website they hand it all over to the criminals, who can then access their genuine account.
4) Phishing emails are often sent to thousands of people, in the hope that someone will read the email and believe its content is legitimate.
5) Many email programs, browsers and firewalls have anti-phishing features that will reduce the number of phishing emails received. There are often giveaways that you can spot, e.g. poor grammar. Emails asking users to follow links or update personal details should always be treated with caution.

SQL Injections give criminals easy access to insecure data

1) Networks which make use of databases are vulnerable to SQL injection attacks.
2) SQL stands for Structured Query Language — it's one of the main coding languages used to access information in databases — see p.80 for more about it.
3) SQL injections are pieces of SQL typed into a website's input box which then reveal sensitive information.
4) A website may allow you to view your account information, as long as you enter your password into an input box. If the website's SQL code does not have strong enough input validation, then someone may be able to enter a piece of SQL code which allows them to access other people's account information as well as their own.

- For example, to access an online retail account you may need to put in a PIN number. When you put in your PIN number, 12345, the website's SQL code may be executed like this:
- SELECT name, address, account number WHERE pin = 12345
- However, this SQL code does not have strong validation because it doesn't specify that the PIN value has to be numerical. This can be exploited by entering the code "12345 OR 1=1". This code is an SQL injection. Now the SQL query looks like this instead...
- SELECT name, address, account number WHERE pin = 12345 OR 1=1
- 1=1 is always true in SQL, so rather than just showing your details, the website instead shows the details of everyone on the website's database.

5) If a website's SQL code is insecure, this can be an easy way for hackers to get past a website's firewall.

No amount of security software can protect against human error...

The best way to prevent social engineering in the workplace is to make employees aware of the dangers — this should be part of a company's network policy, which you'll find more about on the next page. The bottom line is: don't give away any details unless you're sure of who you're giving them to.

Network Security Threats

Organisations use a network policy to protect themselves. A network policy is a set of rules and procedures the organisation will follow to ensure their network is protected against attacks and unauthorised access.

A good network policy will...

- Regularly test the network to find and fix security weaknesses and investigate any problems.
- Use passwords to prevent unauthorised people from accessing the network.
- Enforce user access levels to limit the number of people with access to sensitive information.
- Install anti-malware and firewall software to prevent and destroy malicious software attacks.
- Encrypt sensitive data.

Penetration Testing

- Penetration testing (or pentesting) is when organisations employ specialists to simulate potential attacks on their network.
- Pentesting is used to identify possible weaknesses in a network's security by trying to exploit them. The results of the pentest are then reported back.

Network Forensics

- Network forensics are investigations undertaken to find the cause of attacks on a network. To conduct network forensics, an organisation needs to have a system of capturing data packets as they enter their network.
- After the network is attacked, these packets can be analysed to discover how the network was attacked and decide how to prevent future attacks.

Passwords

- Passwords help prevent unauthorised users accessing the network.
- Passwords should be strong — they should be many characters long, use a combination of letters, numbers and symbols — and be changed regularly.

User Access Levels

- User access levels control which parts of the network different groups of users can access.
- E.g. business managers are likely to have a higher access level allowing them to access more sensitive data, like pay information. They may also have write access to files that others can only read and the ability to change employees' access levels.
- User access levels help limit the number of people with access to important data, so help prevent insider attacks on the network (see p.31).

Anti-Malware

- Anti-malware software is designed to find and stop malware from damaging a network and the devices on it. There are lots of different types of anti-malware software, including antivirus programs which isolate and destroy computer viruses.
- Companies use firewalls to block unauthorised access. Firewalls examine all data entering and leaving the network and block any potential threats.

Encryption

- Encryption is when data is translated into a code which only someone with the correct key can access, meaning unauthorised users cannot read it.
- Encrypted text is called cipher text, while non-encrypted data is called plain text.
- Encryption is essential for sending data over a network securely.

REVISION TASK

Learn these pages and your exam grade should be fairly secure...

Try and find a recent news article about a network attack. Identify the type of attack/malware used. See if you can find out how the attack was carried out and suggest some ways that it may have been prevented. Then try and get over the inevitable fear of ever using the Internet again...

Warm-Up and Worked Exam Questions

That's another Section down. There are lots of little things to remember, like the different protocols and types of malware, so go over them until they're secure in your mind. Then have a go at some questions.

Warm-Up Questions

1) Complete the table showing the names and functions of various network protocols:

Protocol	Function
TCP	
	Responsible for packet switching.
HTTP	
	A more secure version of HTTP.
FTP	
SMTP	
	Used to retrieve emails from a server. The user downloads a copy of the email and the server holds the original email until the user deletes it.
	Used to retrieve emails from a server. The server holds the email until the user downloads it, at which point the server deletes it.

2) Name the type of malware that:
   a) disguises itself as legitimate software.
   b) alters permissions and access levels on the user's device.
   c) spreads by self-replicating without any user help.
   d) encrypts the data on the user's device, making them pay money to the hacker in exchange for the key to decrypt it.

Worked Exam Question

1 Hannah often receives fake emails claiming to be from well-known banks and other organisations.

a) State the name given to the practice of sending fake or spoof emails.

Phishing

[1 mark]

b) Explain the purpose of these fake emails.

They are used to trick people into thinking they are from legitimate organisations so that they give away their personal information, e.g. account login details.

[2 marks]

c) Hannah also receives suspicious emails that contain attachments, sometimes from names in her own contacts list. Explain the dangers of opening untrusted email attachments.

The email attachment could contain a virus. Opening the attachment would activate the virus and cause it to infect the device.

[2 marks]

Exam Questions

2 The Internet offers access to a variety of services, including the World Wide Web.

a) Explain the difference between the Internet and the World Wide Web.

.....

.....

[2 marks]

b) State the function of a Domain Name Server (DNS).

.....

[1 mark]

3 Mahindar sends an email from his smartphone to Holly, who receives it on her laptop.

a) Explain why Mahindar and Holly's devices need IP addresses to connect to the Internet.

.....

.....

[2 marks]

b) Mahindar's email is split into packets and sent over the network using packet switching. Outline what would happen if one of the packets got lost in transit.

.....

.....

.....

Start by thinking about how Holly's laptop would find out that one of the packets was missing.

[3 marks]

4 Kate is a network administrator at a secondary school. She has put measures in place to prevent attacks on the school's network, including having different user access levels.

a) Explain why the school's network needs to have different user access levels.

.....

.....

.....

[3 marks]

b) A hacker recently broke through the school's network security using a brute force attack.

i) Explain what is meant by a brute force attack.

.....

.....

[2 marks]

ii) Identify **two** steps the school can take to protect against a brute force attack.

1. .....

2. .....

[2 marks]

Exam Questions

5 Sally works in an office. Her computer has a MAC address, which helps Sally to access files from the company's server.

a) Describe what is meant by a MAC address.

.....

.....

[2 marks]

b) The network managers at Sally's company work with layers of network protocols.

i) Describe what is meant by a layer of network protocols.

.....

.....

[2 marks]

ii) State the name of **one** layer of network protocols and outline its function.

Layer name: .....

Layer function: .....

[2 marks]

iii) Identify **three** benefits of using layers when working with network protocols.

1. .....

2. .....

3. .....

[3 marks]

6 A magazine publishing company based in rural Scotland connect their computers in a LAN using a Client-Server setup. Their writers live elsewhere in the UK and either email or post their articles to the company, where they are edited in time for the weekly deadline.

Discuss the advantages and disadvantages to the company of changing from their current system to one which uses the cloud.

[6 marks]

Make sure your answer includes both advantages and disadvantages, otherwise you won't be able to get full marks.

7 A law firm has 100 members of staff in an office building in London. The firm stores confidential data about its clients on a server. The firm currently has no network policy.

Discuss how a network policy could benefit the law firm.

[8 marks]

Consider the threats posed to the firm's network and how a network policy could protect against them.

Revision Questions for Section Two

From passwords to packets to peer-to-peer to protocols — this section has been quite the mouthful...

- Try these questions and tick off each one when you get it right.
- When you've done all the questions for a topic and are completely happy with it, tick off the topic.

LANs, WANs and Hardware (p.20-22) ☐

1) What's the difference between a LAN and a WAN? ☐
2) Give three factors that can affect the performance of a network. ☐
3) Give one similarity and one difference between a switch and a router. ☐
4) Give one advantage of using wired network connections over wireless. ☐
5) What's the difference between a Wireless Access Point (WAP) and a hotspot? ☐
6) What are Wi-Fi® channels and how are they used? ☐

Network Types and Topologies (p.23-24) ☐

7) Draw diagrams of a Client-Server network and a Peer-to-Peer (P2P) network. ☐
8) Compare the client-server relationship with the relationship of peers in a P2P network. ☐
9) Give two reasons why someone might choose to set up a peer-to-peer network. ☐
10) Give three advantages and two disadvantages of using a star network. ☐
11) Describe the key features of a mesh network. ☐

Protocols and The Internet (p.27-30) ☐

12) What is the definition of a protocol? ☐
13) What is the difference between a MAC address and an IP address? ☐
14) List three things that a data packet contains. ☐
15) How does a receiving device know whether a packet has been corrupted or has gone missing? ☐
16) What does each of the following stand for? Describe in a sentence what each one does:
TCP, IP, FTP, HTTP, HTTPS, SMTP, POP3, IMAP ☐
17) List the 4 layers of network protocols. ☐
18) Give three reasons why we divide protocols into layers. ☐
19) Describe how a Domain Name Server (DNS) works. ☐
20) Give five advantages and five disadvantages of using the cloud. ☐
21) What is a virtual network? ☐

Network Security Threats (p.31-33) ☐

22) Describe, in a sentence each, five different types of network attack. ☐
23) List six actions that malicious software might carry out. ☐
24) Describe three ways that malware can access a device or network. ☐
25) What is social engineering? Give two examples of it. ☐
26) Explain how an SQL injection works. ☐
27) Give five security measures a good network policy might cover. ☐
28) Give three precautions you should take with your passwords. ☐
29) What do organisations use firewalls for? ☐

Ethical and Cultural Issues

Despite what you might think, computer science doesn't just exist in a well-ventilated bubble — it affects all of our lives. Computers, new technology and the Internet all impact different people in different ways.

Use of Technology can raise all sorts of Tricky Issues

1) Ethical issues are about what would be considered right and wrong by society.
2) Legal issues are about what's actually right and wrong in the eyes of the law.
3) Cultural issues are about how groups of people with particular beliefs, practices or languages may be affected, e.g. ethnic groups, religions, countries.
4) Environmental issues are about how we impact the natural world.

These categories will often overlap — many environmental and cultural questions could also be considered questions of ethics

If a company acted legally but ignored all questions of ethics, it could lose public trust. Many companies have a code of conduct (a set of rules that the company and its employees will follow) to show that they take these issues seriously. A company may invent its own code or agree to follow a standard one.

New technologies affect Different Stakeholders

1) Stakeholders are individuals or groups of people who have an interest in or are affected by a particular scenario (e.g. the actions of a company or the development of a new technology.)
2) Stakeholders may include a company's owners, its employees, the shop that sells the company's product, customers, the company's hardware suppliers and the local community.
3) Each group of stakeholders has different priorities which may conflict with those of the others.
4) In the exam, you may be given a scenario and asked to discuss the ethical, legal, cultural or environmental issues it raises for the various stakeholders.

EXAMPLE:

Sally pays to download movies from a major on-demand streaming service. Her friend suggests that she should use a website where the movies are free but that probably isn't legal. Identify the key stakeholders and discuss the ethical and legal issues this raises for each. ***[8 marks]***

Sally is a stakeholder who faces ethical and legal dilemmas. If she uses the website, she would save money, but may be supporting copyright theft and, indirectly, other criminal activities. She may also be breaking the law herself and could face prosecution. By using a website of dubious nature, she will put herself at risk from computer viruses and other malware. She may argue that the website is easily accessible, the legality is a grey area and that the government or film company should shut down these websites if they don't want people to use them.

The movie's creators, publishers and employees are stakeholders. They may lose money as a result of people using these websites. This could affect their ability to employ staff and make films in the future. They could use copyright laws, e.g. the Copyright, Design and Patents Act to attempt to prosecute the website's owners or the website's users. If they targeted users, this could be an effective scare tactic, but could also create bad press for the company.

Owners of the legitimate streaming service are also stakeholders who would lose money which could affect their ability to employ staff or stay in business.

Governments could also be stakeholders. The film company's government will want to protect its companies and will be under pressure to take action. However this may be very expensive and technically difficult. If the website is hosted in a different country with different copyright laws, tricky political negotiations may be required. The government may also be concerned that money from these websites could be funding further illegal activities.

EXAM TIP

These are really common essay questions in the exam...

When you're answering questions on these issues, get in the habit of thinking about what groups are affected (stakeholders) and what issues (legal/ethical/cultural/environmental) it raises.

Ethical and Cultural Issues

It's quite concerning to think how many people have access to your social media profile. However, some people in the world don't get to access social media at all — so maybe you're the lucky one...

It's hard to keep information **Private** on the Internet

1) Many websites (e.g. social media, banking and retail) require users to provide personal information in order to set up an account, e.g. date of birth and address.
2) Social media websites actively encourage you to post even more personal information, including photographs and details of your job and social life.
3) Cloud computing websites allow users to upload personal files to their servers.
4) Companies may make your personal information, photos, etc. available to other website users or the whole Internet. They may also sell your personal details, buying habits, likes / dislikes etc. to other organisations (who might use it to send you targeted adverts or spam emails). Companies can do lots with your information as long as they stay within the bounds of the privacy agreement.
5) Users will accept a privacy agreement before using many websites and software. The trouble is that very few people actually read these so are unaware of what they're agreeing to. Even if they do read the terms, users often have no choice but to agree if they want to use the website or software at all.
6) Users can take steps to make the information they share more private, e.g. change their privacy settings on social media sites. Websites often have fairly relaxed privacy settings by default. They also can have privacy settings that are hard to find and hard to understand.
7) Users have to trust companies to keep their data secure. But this doesn't always happen — there have been various high profile cases where customer data held by large companies has been leaked or stolen.
8) There are also issues around having so much personal information accessed via mobile devices, which might be stolen.

Censorship and **Surveillance** are controversial issues

Censorship

1) Internet censorship is when someone tries to control what other people can access on the Internet. Some countries' governments use censorship to restrict access to certain information.
2) One of the strictest countries for censorship is China, where they restrict access to websites which are critical of the government. China also censors many major foreign websites, including Facebook®, Twitter and YouTube™. In Cuba, citizens can only access the Internet from government-controlled access points.
3) Many governments use some form of censorship. Many countries (including the UK) restrict access to pornography, gambling and other inappropriate websites in order to protect children.

Surveillance

1) Computer surveillance is when someone monitors what other people are accessing on the Internet.
2) Many countries use some form of surveillance. Government intelligence agencies may use packet sniffers and other software to monitor internet traffic, looking out for certain key words or phrases that might alert them to illegal activities, terrorism, etc. In some countries Internet Service Providers (ISPs) keep records of all websites visited by all its customers for a certain amount of time.

Censorship and surveillance are controversial topics. Some people support them in some form, e.g. to protect children or to stop terrorism. Others are completely against them, including several non-profit organisations which campaign against what they call cyber censorship and mass surveillance.

You should have seen this page before it was censored...

The weird thing about this stuff is that everyone knows it's happening but no-one does anything about it — the Internet is such a big part of modern life that for many people their loss of privacy is a price worth paying. Other people aren't so keen about losing their privacy though, which is why these issues are so controversial.

Ethical and Cultural Issues

This page might talk about trolls, but there's nothing fantastical or whimsical about it.
In fact, this page gets into some serious issues which have damaged people's lives.

New technology can impact our **Social Well-being**

1) Companies release new technology regularly, and pay for advertisements to promote it. These techniques often try to influence and pressure people into buying or upgrading to the latest device.
2) Technology has also increased peer pressure — children feel pressure to own the latest device for fear of being bullied by their peers. Parents feel pressured into buying them.
3) Smartphones make it easier for people's work to intrude into other areas of life. Employees may be expected to carry a smartphone all the time, so that they can always be contacted — the smartphone may beep each time they get a work e-mail. This can be stressful for employees who feel they can never really switch off from work.
4) Face-to-face social interaction can be neglected as more of our social lives move online. This is made worse by having the Internet on mobile devices — it's now almost possible to ignore real life completely.

Cyberbullying and **Trolling** are a problem on **Social Media**

Cyberbullying

- Cyberbullying is when somebody uses social media to deliberately harm someone else.
- This includes trying to intimidate or insult someone, or trying to humiliate or defame them (damage their reputation).
- Cyberbullying can cause serious distress — people have been driven to suicide because of these attacks.

Trolling

- Trolling is when somebody tries to cause public arguments with others online.
- For example, the troll may take part in a political discussion online, but only to make comments which would frustrate the other members of the discussion. Trolls normally do this for their own amusement or to gain attention.

1) Problems like cyberbullying and trolling may be a result of the anonymity that the Internet gives people. They say things online that they wouldn't say if talking to someone face-to-face.
2) The Internet has made it easier for children to access inappropriate material, like pornography, drugs and gambling. Parents and schools can use parental-control software to try to stop children seeing it.
3) Sexting (sending sexually explicit messages or images to other people) is more common as smartphones and video messaging applications have become more popular. Sexting can be dangerous as the person receiving the images might not be trustworthy — social media allows them to forward someone else's images onto anyone they want. There are now laws which try to prevent this.

Technology has enabled a lot of anti-social behaviour...

These are all sensitive topics, but they're worth mentioning in the exam if they're relevant to the question. The problems on this page can have a damaging impact on the mental wellbeing of the victim — if you ever experience any of them, the best thing to do is to tell someone about it.

Ethical and Cultural Issues

Overuse of technology isn't just dangerous to mental health — it can also cause physical conditions. But as it becomes more and more central to our lives, these problems can be harder to avoid...

Using technology too much can cause **Health Problems**

1) Eyestrain can be caused by looking at a device's screen for too long. This is a particular problem if the device is used in bad lighting, the screen is flickering or there is sunlight glare on the screen.

Eyestrain can be prevented by using suitable lighting, keeping the screen a good distance away from your eyes and taking regular breaks from using your device.

2) Repetitive Strain Injury (RSI) is when parts of the body (normally fingers and wrists) become damaged as a result of repeated movements over a long period time, such as typing on a keyboard.

RSI can be prevented by having a correct posture, arranging your desk appropriately and taking regular breaks.

3) Sitting at a computer too long can cause back problems. Back pains are normally caused by poor posture.

You can prevent back pains by using an adjustable chair, a foot rest and/or an adjustable monitor, and ensuring that you aren't sitting at an awkward angle.

Technology and the Internet have shaped our **Culture**

Selfies (photos we take of ourselves) have become really popular because smartphone cameras and social media allow us to take them and share them easily. But could they be seen as a sign that social media is gradually making people more attention-seeking and self-obsessed...

Viral is a word used to describe videos, images or messages on the Internet which have rapidly spread over social media and have been seen by millions of people. Companies, politicians, celebrities and charities all try to use images and videos in their promotional campaigns in the hope that they will go viral.

Social media and blogging websites allow people to publish writing, art or other media. This can give a voice to groups of people who might have been ignored by mainstream media.

There are benefits and problems with all technologies...

Pick a new technology that you know quite a lot about. Think about what opportunities and conveniences it can give to its users. Try and come up with some issues that it can cause as well — think about the effects of overusing it, or of people abusing it. Are there any other stakeholders, other than the company behind it and the end user? What issues do they face?

Ethical and Cultural Issues

Computers have had a tremendous impact on our culture — it's difficult to imagine some of the things we do today without them. Just remember that many people have been left behind by this technological revolution.

New technology is changing how we do **Business**

1) Music and television streaming services have allowed their customers to listen and watch media for less money, usually through a subscription service. But some people aren't happy about it — e.g. musicians who feel streaming companies don't pay them enough money to use their music.
2) The sharing economy is the name given to services which use new technology to let people make money from things they already own — e.g. Uber™ lets you turn your car into a taxi service, and Airbnb™ uses the Internet to let you rent out a room in your house to tourists.
3) These services are cheap, but they draw customers away from taxi firms and hotel owners. Also, they may be more risky for sharers and customers. E.g. the sharer may not know the safety regulations they should follow, and may find their insurance policy won't cover them if there's damage or theft.

Unequal access to technology has caused a **Digital Divide**

1) The digital divide is created by the fact that some people have greater access to technology than others. E.g. people can use the Internet to apply for jobs or university courses, access a range of services from banking to retail, and keep in touch with friends. People who have a limited access to the Internet are therefore at a heavy disadvantage.

CAUSES OF THE DIGITAL DIVIDE

- Some people don't have enough money to buy new devices like smartphones and laptops, which can be very expensive.
- Urban areas are likely to have greater network coverage than rural areas.
- Some people don't know how to use the Internet and other new technologies, and so are shut out of the opportunities they offer. This is a problem for many older people who haven't grown up with computers and so have little experience with them.

2) The global divide is created by the fact that the level of access to technology is different in different countries. People in richer countries tend to have greater access to technology than people in poorer countries. The Internet and other technologies have created lots of opportunities for the people with access to them, so this has increased the inequality between poorer and richer countries.
3) Projects have been set up to combat the digital and global divides. There are several British community projects aimed at improving Internet coverage in rural areas. One Laptop Per Child is a charity which provides laptops to children in Africa, Central Asia and South America.

These issues have got my fingers arguing — it's a digital divide...

Technology doesn't just affect those who use it — it affects everyone, including the people who can't use it. In the exam, think about how new technology might contribute to the digital divide, as well as all the other issues covered in this section. There's often loads you can say, so don't focus too much on one single issue.

Warm-Up and Worked Exam Questions

You're probably already aware of some of the issues on these pages, but it's worth going back over any that are new to you. The more you can remember, the more marks you'll be able to bank in the exam.

Warm-Up Questions

1) Tick one box in each row to show whether it concerns censorship or surveillance.

	Censorship	Surveillance
A business monitors what their employees view online.		
A country's government blocks access to Facebook®.		
A government agency intercepts emails containing certain words.		
A school restricts access to harmful websites.		
An Internet Service Provider collects data on browsing habits.		

2) Give an example of a physical health risk caused by extended use of technology, and suggest one way that it could be prevented.

3) What is meant by the 'digital divide'?

Worked Exam Question

1 Jasmine uses several social media websites. She was recently a victim of cyberbullying.

a) Define the terms cyberbullying and trolling.

Cyberbullying: The use of social media to deliberately harm someone else.

Trolling: Causing arguments or provoking anger and frustration online.

[2 marks]

b) Suggest **two** reasons why cyberbullying and trolling have become a problem on social media.

1. Social media can give people greater anonymity than they would have in real life.

2. There is often no punishment if somebody behaves badly online.

[2 marks]

c) Social media has also had a positive impact on how we communicate. Describe **one** way in which social media benefits its users.

Social media allows friends to keep in touch over a long distance, which helps maintain and even improve our social lives.

It also allows potentially cheap communication, and easy sharing of media such as photos

[2 marks]

Exam Questions

2 A supermarket replaces all of their staffed checkouts with electronic self-service checkouts. This increases the company's profits, but causes some of the checkout workers to lose their jobs.

a) Define the term stakeholder.

.....

[1 mark]

b) Identify **two** stakeholders in the decision to replace the staffed checkouts. State whether the effect on these stakeholders was positive or negative and give a reason for your answer.

Stakeholder 1: .....

The effect on this stakeholder was: Positive / Negative

Reason: .....

Stakeholder 2: .....

The effect on this stakeholder was: Positive / Negative

Reason: .....

[6 marks]

3 Tom works for a smartphone company. He is stressed as he feels he can't switch off from work.

a) Explain how new technology could have allowed work to intrude into other areas of Tom's life.

.....

.....

[2 marks]

b) Tom's company releases a new smartphone every year. Outline the social pressures that can be created by the regular release of new technology.

.....

.....

[2 marks]

c) Tom's neighbour, Jerry, is also stressed because he is finding it difficult to apply for jobs. All the companies he wants to apply to only accept online applications, and he has limited access to the Internet. Explain **one** reason why a person's Internet access might be limited.

.....

.....

[2 marks]

4 In many factories robots have replaced humans for routine tasks such as cutting and joining materials together and retrieving products stored in a warehouse.

Discuss the impact of robots replacing humans to carry out routine tasks in factories. In your answer you might consider: stakeholders, technology and ethical issues.

Try to talk about both positive and negative effects that the robots might have.

[6 marks]

Environmental Issues

Devices all have an environmental impact. Take a smartphone — it's made of materials that are mined from the Earth, it consumes energy when used, and when it's thrown away it could end up on a landfill site.

When we **Make** devices we use up **Natural Resources**

1) Electronic devices contain lots of raw materials.
2) Plastics (which are used for casing and other parts) come from crude oil.
3) Devices also contain many precious metals like gold, silver, copper, mercury, palladium, platinum and indium. Many of these metals only occur naturally in tiny quantities.
4) Extracting these materials uses lots of energy, creates pollution and depletes scarce natural resources.

When we **Use** devices we use **Energy**... lots of it

All the billions of devices in the world today are consuming energy in the form of electricity — a lot of it.

1) Most electricity is made using non-renewable resources like coal, oil and gas. Extracting these resources and producing electricity in power stations causes lots of pollution including greenhouse gases.
2) All computers generate heat and require cooling. The powerful servers used by businesses and the Internet are a particular problem. They're very power hungry and require special air-conditioned rooms to keep them cool. That means using even more energy and more pollution.
3) Devices also waste a lot of energy. Servers normally only use a small proportion of their processing power. People often leave their desktops, laptops and smartphones idle. This means these devices are using a lot of energy without actually doing anything.
4) There are several ways to reduce the amount of energy wasted by devices:

- Virtual servers are software-based servers rather than real machines. Multiple virtual servers can run on one physical server, so the physical server can run at full capacity.
- Most modern devices include sleep and hibernation modes to reduce their power consumption when they are idle.
- Don't leave electronic devices (TVs, laptops, etc.) on standby.

When we **Throw Away** devices we create loads of **E-waste**

1) E-waste is a huge problem — the world creates 20-50 million tonnes of e-waste every year. Modern devices have a very short life before they're discarded — either because they break or because people want to upgrade.
2) Device manufacturers and retailers are part of this problem. They provide short warranties (e.g. 1 year), use marketing to convince people to upgrade and have pricing policies that make it cheaper to replace than to repair.
3) The Waste Electric and Electronic Equipment (WEEE) directive was created to tackle the e-waste problem. The WEEE has rules for disposing of e-waste safely, to promote reuse (e.g. refurbishing broken devices to use again) and recycling (e.g. extracting the devices' precious metals).
4) To cut costs a lot of e-waste is sent to certain African and Asian countries where regulations are less strict. Here, most of it ends up in landfill and can be a hazard — toxic chemicals can leak into the ground water and harm wildlife.

Don't (e-)waste your time — use your energy to learn this page...

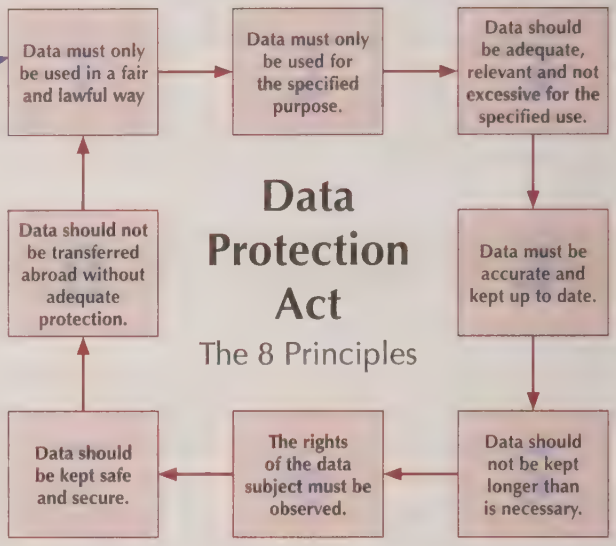
From manufacture right through to when they're thrown away, our devices put a strain on the environment. But it's not all bad — they let us communicate without having to travel long distances in pollution-spouting vehicles, and reduce our need for paper. However, whether these make enough of a difference is debatable.

Computer Legislation

There are now lots of different laws related to computing, but fortunately you only have to learn about a select few. There's a lot of (freedom of) information here, so make sure you take your time to go over it.

The Data Protection Act controls the use of personal data

1) The Data Protection Act 1998 gives rights to data subjects (people whose personal data is stored on computer systems). The Act has eight principles, as shown here.
2) Before collecting personal data an organisation must register with the government, saying what data they'll collect and how they'll use it.
3) The Act gives data subjects the right to see the personal data an organisation holds about them.
4) There are some exceptions to this, e.g. organisations don't have to disclose the data they hold if it could affect national security, tax assessment, or the outcome of a court case.
5) If a data subject feels an organisation's use of their data has caused them distress, they may be entitled to compensation.



The Freedom of Information Act allows public access to data

1) The Freedom of Information Act 2000 allows members of the public to access information held by a public organisation about that organisation's activities. The Act covers information stored in computer data files (including audio and video files), emails, and printed documents.
2) Public organisations include government departments, the Houses of Parliament, local councils, the armed forces, the National Health Service, police authorities, schools and universities.
3) The Act makes public organisations publish certain information on a regular basis so that the public have access to it. It also allows members of the public to request specific information.
4) There are some exceptions to the Act. E.g. an organisation can withhold requested information if it is intended for future publication, or if disclosing it could affect national security or cause people harm.

The Computer Misuse Act prevents illegal access to files

The Computer Misuse Act 1990 was introduced to stop hacking and cyber crime. It introduced three new offences:

1) Gaining unauthorised access to a private network or device, e.g. through hacking (just accessing a network could get you a fine or prison sentence).
2) Gaining unauthorised access to a network or device in order to commit a crime, like stealing data or destroying the network.
3) Unauthorised modification of computer material — e.g. deleting or changing files. The Act also makes it illegal to make, supply or obtain malware.

REVISION TIP

Some of these laws are a bit similar — don't mix them up...

The Freedom of Information Act deals with information held by organisations about themselves, and the Data Protection Act deals with people's personal data held by organisations. The Computer Misuse Act hasn't changed much since 1990, but it has been amended to give harsher punishments to offenders, and to give more power to law enforcement agencies, e.g. to carry out surveillance.

Computer Legislation

Another part of computer legislation covers intellectual property — stuff people create. Your favourite song, the essay you wrote and even this book are intellectual property, and these laws are here to help protect it.

The Copyright, Designs and Patents Act protects innovation

1) The Copyright, Designs and Patents Act 1988 was introduced to protect intellectual property — anything someone has created, e.g. a novel, a song, piece of software, a new invention.
2) Copyright covers written or recorded content, e.g. books, music, films, software, video games.
3) The Act makes it illegal to share copyrighted files without the copyright holder's permission, use unlicensed software or plagiarise (copy) somebody else's work. Copyright holders can make money by granting permission to use the material for a fee.
4) Patents cover new inventions — they protect ideas and concepts rather than actual content. E.g. if a company invented a new method of charging smartphone batteries, they could obtain a patent to prevent other companies from releasing smartphones that used this process. In computing, patents mostly apply to pieces of hardware.
5) The Internet has made it harder to protect copyrighted content due to the ease of file sharing. It's also difficult to enforce copyright if content is held on servers in countries with more relaxed copyright laws.
6) A lot of illegal file sharing takes place over peer-to-peer networks (p.23) using the BitTorrent® protocol to share files directly between devices. Cloud-based (p.30) file-hosting websites are also used — copyrighted content is uploaded to the website where anyone with an account can download it.
7) It's a grey area as to how much responsibility the website owners have for content that users upload. However many of the most popular Torrent and file-hosting websites used for illegal sharing have eventually been prosecuted for copyright violation and forced to shut down.

Creative Commons licences allow legal file sharing

1) Creative Commons (CC) licences allow you to legally share media and software online without having to ask for permission first. Intellectual property owners use creative commons licences when they want other people to share or build upon their work.
2) There are four main types of creative commons licence:

Type of CC licence	Conditions
Attribution	Work can be shared, copied or modified, but the copyright holder has to be credited.
Share-alike	Modified works can only be distributed with the same license terms as the original.
Non-commercial	Nobody can use the copyrighted work for profit.
No derivative works	The work can be copied and distributed, but can't be modified or built upon.

3) CC licences are often combined, e.g. attribution share-alike or attribution share-alike non-commercial.
4) Some works are in the public domain — they don't have any copyright attached to them, meaning you can copy and share them as you wish. UK copyright expires 70 years after the creator's death, at which point their creation enters the public domain — Shakespeare's plays, Beethoven's symphonies and da Vinci's paintings are all now in the public domain.

Intellectual property covers loads of different media...

Search the internet for something that has a Creative Commons licence. What kind of licence is it? What benefits does the licence give the community? What about the creator? What problems might it cause that a stricter copyright licence would avoid?

Warm-Up and Worked Exam Questions

Make sure you know those laws inside out — it'd be easy to mix them up and drop some marks. Once you're confident with the last few pages, try out these questions — then go back over anything that slips you up.

Warm-Up Questions

1) Which of the following are materials used to create electronic devices?
copper jupiter platinum plastic cedar wood potato mercury mithril gold silver

2) State which law the following issues relate to:
   a) A hospital holds the medical records of its patients so they can be treated.
   b) A criminal hacks into a broadband company's network and steals its customers' account details.
   c) A request is made to a university to release information regarding the amount their vice chancellor is paid.
   d) A polling company holds data on members of the public for a survey it is conducting.
   e) An employee accesses their manager's network account and deletes company data.

3) Define what each of these Creative Commons licence conditions means for the intellectual property that is being shared:
   a) Attribution
   b) Non-commercial
   c) Share-alike
   d) No derivative works

Worked Exam Questions

1 The average household can spend almost £100 a year on wasted electricity.

a) Identify **two** ways that electronic devices waste electricity.

1. Users sometimes leave devices idle / on standby.
2. Devices generate excess heat due to inefficiency.

[2 marks]

b) Explain how hardware manufacturers can limit the amount of electricity wasted by devices.

Manufacturers can include sleep or hibernation modes in new devices to reduce their energy consumption when they are idle.

[2 marks]

2 The Data Protection Act 1998 gives rights to data subjects.

a) Define the term data subject.

Someone whose personal data is stored on somebody else's computer system.

[1 mark]

b) State **two** principles of the Data Protection Act.

Any of the eight principles would be fine here.

1. Data must only be used in a fair and lawful way.
2. Data should be kept safe and secure.

[2 marks]

Exam Questions

3 Illegal file sharers infringe upon the copyright of musicians and film makers.

a) Define the term copyright.

.....

[1 mark]

b) Some pieces of music are in the public domain. Explain what is meant by the public domain.

.....

.....

[2 marks]

4 The average smartphone is only used for two years before it is discarded as e-waste.

a) Define the term e-waste.

.....

[1 mark]

b) Identify **two** reasons why smartphones are only used for a short amount of time before they are discarded.

1. .....

2. .....

[2 marks]

c) Outline the problems that the short life span of electronic devices has on the environment.

.....

.....

.....

.....

[4 marks]

d) Outline how the problem of e-waste can be managed to limit its impact on the environment.

.....

.....

[2 marks]

5 A cinema collects information from customers who book seats to watch movies.
The cinema would like to store this information for the following reasons:

- to make it easier for customers when they book seats in the future.
- to enable the cinema to contact customers with details of future films.

Explain the measures that the cinema should take to ensure that customer data is stored legally.

Think about what legislation they might be at risk of breaking, and what they can do to avoid this. *[6 marks]*

Revision Questions for Section Three

Well, that section had a lot of issues — thankfully you're not here to solve its problems, just learn its content.

- Try these questions and tick off each one when you get it right.
- When you've done all the questions for a topic and are completely happy with it, tick off the topic.

Ethical and Cultural Issues (p.38-42) ☐

1) Define each type of issue in a sentence: ethical, legal, cultural, environmental ☐
2) What is a stakeholder? Identify five stakeholders in the release of a new smart phone. ☐
3) Give two reasons why someone might give their personal details to a website. ☐
4) Give two problems with many online companies' privacy agreements. ☐
5) What can you do to make the information you share online more private? ☐
6) Explain the difference between censorship and surveillance. ☐
7) Give one argument for and one against Internet censorship. ☐
8) Give one argument for and one against governments carrying out Internet surveillance. ☐
9) Give four examples of how new technology may affect social well-being. ☐
10) What is cyberbullying? ☐
11) What is an Internet troll? ☐
12) Give two reasons why cyberbullying and trolling have become so common. ☐
13) What is sexting and why is it dangerous? ☐
14) Give three examples of health problems which can be caused by using a computer. ☐
15) Give three examples of how technology and the Internet have shaped our culture. ☐
16) Give three reasons for why a digital divide exists. ☐

Environmental Issues (p.45) ☐

17) Give three examples of natural resources which are used to make computers. ☐
18) Explain how a device's need for energy impacts the environment. ☐
19) Give three ways to reduce the amount of energy devices waste. ☐
20) What is e-waste and why do we generate a lot of it? ☐
21) Describe an environmental danger caused by e-waste left in landfill sites. ☐

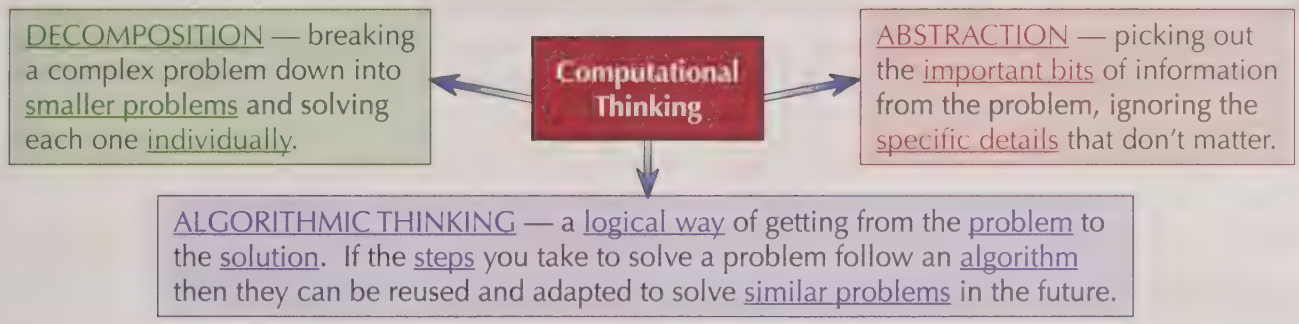
Computer Legislation (p.46-47) ☐

22) What are the eight principles of the Data Protection Act 1998? ☐
23) a) Which Act allows the public to see information held by a public organisation about its activities?
    b) Give two exceptions to this Act that allow organisations to withhold information. ☐
24) What were the three new offences introduced by the Computer Misuse Act 1990? ☐
25) What is intellectual property? ☐
26) Why do we use copyright? ☐
27) Give three things that the Copyright, Designs and Patents Act 1988 makes illegal. ☐
28) Give the four types of Creative Commons licence. ☐
29) Why might a creator of intellectual property want to use a Creative Commons licence? ☐

Computational Thinking

Computational thinking is all about the steps you take to find the best solution to a complex problem.

Three Key Techniques for Computational Thinking



These techniques are all used in Real-Life...

Computational thinking is something you'll do all the time without even noticing.

For example, when deciding which film to watch at the cinema with your family:

Decomposition	Abstraction	
Things to look at	Details to ignore	Details to focus on
What type of films are on?	Plot details, actors and director.	Film genre and age rating.
What times are the films on?	Days other than the date you're going.	Start and end times on the date you're going.
What are the reviews like?	In depth analysis of the characters and plot.	Ratings

Algorithmic thinking may involve coming up with some logical steps to reach a decision. E.g. listing all of the films that are showing, then deleting all the age restricted films and ones with poor ratings. Getting each family member to vote for their favourite, then picking the film with the most votes.

If the family went to see a film the following week they could use the same processes of decomposition, abstraction and algorithmic thinking, but they would have to do the research and make the decisions again.

... and the Same Skills can be used in Computer Science

Computer scientists rely on decomposition, abstraction and algorithmic thinking to help them turn a complex problem into small problems that a computer can help them to solve.

Imagine the task is to sort a list of product names into alphabetical order:

See p.57-59 for more on sorting algorithms.

- One part of the decomposition might decide what alphabetical order means — letters are straightforward but what if some entries in the list contain numbers and punctuation?
- Another part of the decomposition might look at comparing the entries — this could be decomposed further into how you could compare two entries, three entries, etc.
- Abstraction will help the programmer focus on the important bits — it doesn't matter what the entries are and what they mean. The important information is the order of the characters in each entry.
- Algorithmic thinking will put the tasks into a step by step process. For example, you might compare the first two entries and order them, then compare the third entry to each of the first two and put it in the correct place, then compare the fourth entry to each of the first three, etc.

Break your big problems down into small manageable tasks...

Think about a recent decision you've made, or a problem you've solved. How did you decompose the problem? What information did you ignore/focus on? How did you reach the final solution?

Writing Algorithms — Pseudocode

Algorithms are just sets of instructions for solving a problem. In real-life they can take the forms of recipes, assembly instructions, directions, etc. but in computer science they are often written in pseudocode.

Algorithms can be written using Pseudocode

1) Pseudocode is not an actual programming language but it should follow a similar structure and read like one (roughly). The idea is that pseudocode clearly shows an algorithm's steps without worrying about the finer details (syntax) of any particular programming language.
2) It is quick to write and can be easily converted into any programming language.
3) There are different ways to write pseudocode — they are all equally correct as long as the person reading the code can follow it and understand what you mean.

EXAMPLE: **Write an algorithm using pseudocode to calculate the salary of a worker after a 10% pay increase.**

A simple solution to the problem would be:

```
Take worker's current salary
Multiply the salary by 1.1
Display the answer
```

This solution is perfectly adequate as the problem has been split down into steps and it is obvious to the reader what to do at each stage.

A more useful solution is shown here:

```
int salary
salary = input("Enter your salary.")
newsalary = salary * 1.1
print(newsalary)
```

This solution is better as the words and structure resemble a real programming language. It can be more easily adapted into real code.

Make sure your pseudocode isn't Too Vague

Even though pseudocode isn't a formal programming language you still need to make sure it's readable, easy to interpret and not too vague.

EXAMPLE: **When registering on a website, a user's password should be more than 6 characters long and it must be different from their username. Write an algorithm to check if the password is valid. If it's invalid it should say why.**

```
IF the length of the password is less than or equal
to 6 characters long OR password is the same as the
username THEN it is invalid ELSE the password is valid
```

This code is too vague and unstructured. It won't give reasons why the password is invalid and doesn't give any input variables (see p.66).

The pseudocode asks the user to input a username and password and stores them as variables.

The code gives different outputs depending on why the password is invalid.

The indentation of the pseudocode makes it more readable.

```
username = input("Enter your username.")
password = input("Enter your password.")
if length of password <= 6 then
    print("Password is too short.")
else
    if password == username then
        print("Password is the same as username.")
    else
        print("Password is valid.")
    endif
endif
```

The first IF statement checks to see if the password is too short and the second checks if it's the same as the username.

See p.70 for more on IF statements.

EXAM TIP

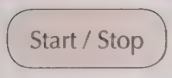
Pseudocode isn't always everything it appears to be...

If you have to write an algorithm in your exam, pseudocode is a great way to give your answer. You don't have to worry about the fiddly bits of syntax from a specific programming language.

Writing Algorithms — Flow Diagrams

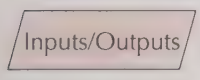
Algorithms can also be shown using a flow diagram, and just like for pseudocode, there are different ways to write the same algorithm. You do get to draw some different shapes though, so things are looking up.

Flow diagrams use **Different Boxes** for different **Commands**

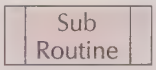


The beginning and the end of the algorithm are put in boxes with rounded corners.

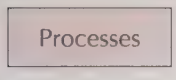
Decisions, often a 'yes' or 'no' question, are put in diamond boxes.



Anything that's put into or taken out of the algorithm goes in a parallelogram box.



Sub Routines are like sub programs (p.81-82) — they reference other flow diagrams.



General instructions, processes and calculations go in rectangular boxes.

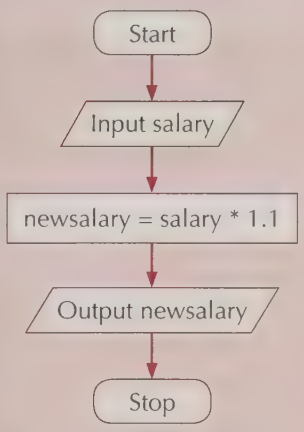
Arrows connect boxes and show the direction you should follow. Some boxes might have multiple arrows coming in or going out of them.

Algorithms can be written as **Flow Diagrams**

Flow diagrams can show sequences, selections, iterations or a combination of them.

Sequences

A flow diagram for calculating a salary after a 10% increase.

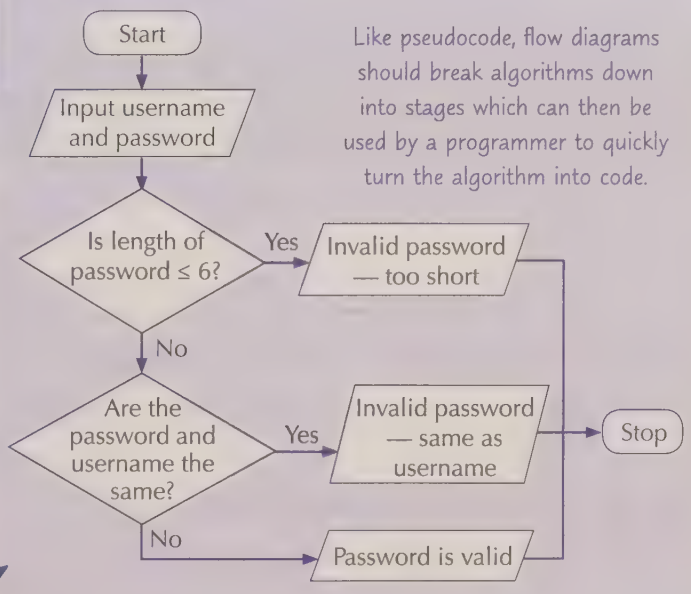


This flow diagram shows a sequence. There is only one way from the start to the end.

The pseudocode for these two algorithms is on the previous page.

Selections

A flow diagram for checking that a password is more than 6 characters long and different from the username.

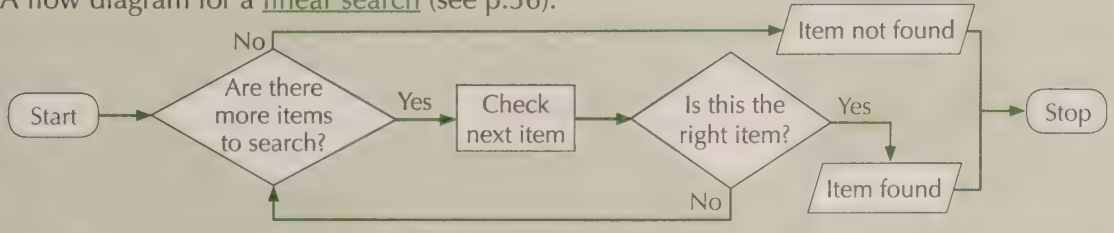


Like pseudocode, flow diagrams should break algorithms down into stages which can then be used by a programmer to quickly turn the algorithm into code.

This flow diagram shows selections. There are multiple ways to get from start to stop.

Iterations

A flow diagram for a linear search (see p.56).



This flow diagram shows an iteration — it contains a loop that allows you to repeat a task.

Flow diagrams should show the general flow of the algorithm...

You don't need to pack flow diagrams with all the details. If you get a flow diagram question in your exam, make sure you use the correct boxes and that all paths in your diagram lead to the end.

Warm-Up and Worked Exam Questions

Now it's time to practise your computational thinking, pseudocode and flow diagram skills.
Work through these warm-up questions then have a go at the exam questions.

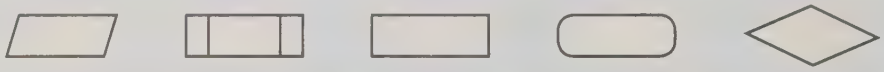
Warm-Up Questions

1) Give the names of the three key techniques used for computational thinking.
2) What is meant by an 'algorithm'?
3) Which of these statements are true?

A: Pseudocode is a formal programming language.
B: There are lots of different ways to write pseudocode.
C: The more vague that pseudocode is the better.
D: Indentation helps to make pseudocode easier to follow.

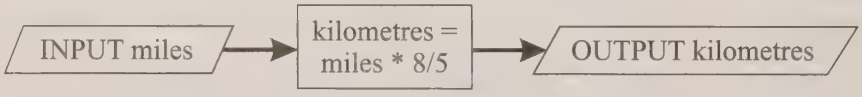
4) Draw lines matching the commands below to the correct flow diagram symbol.

Start Output Decision Sub routine Process



Worked Exam Question

1 The flow diagram below shows how to convert miles to kilometres.



a) Identify **one** problem with this flow diagram.

Think about what is missing from the flow diagram.

There are no boxes for starting and stopping.

[1 mark]

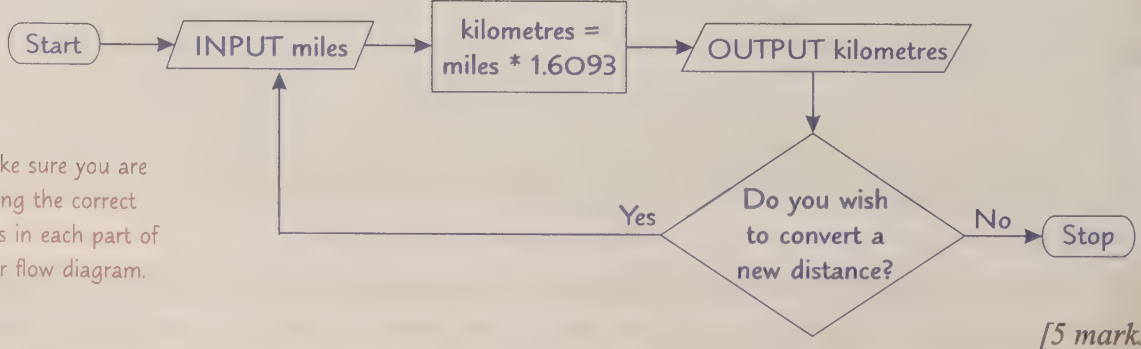
b) State the distance in kilometres of 10 miles.

kilometres = 10 × 8 ÷ 5 = 16

16 km

[1 mark]

c) Draw an improved flow diagram by:
- Using a more accurate conversion factor of 1.6093 instead of 8/5.
- Asking the user if they wish to convert another distance.
  If yes, the flow diagram performs the new calculation and if not, the flow diagram ends.



Make sure you are using the correct boxes in each part of your flow diagram.

[5 marks]

Exam Questions

2 Bernard has written the algorithm on the right using pseudocode.

```
height = input("Enter the height.")
width = input("Enter the width.")
area = height * width
print(area)
```

a) Describe what Bernard's algorithm does.

Describe what is happening on each line

.....

.....

.....

[3 marks]

b) State the value of `area` if `height = 5` and `width = 10`.

.....

[1 mark]

3 A file uploading service won't allow two files with the same file name to be uploaded. If a file name already exists, it will ask the user to change the file name.

a) Describe with examples how abstraction can help decide how to compare the files.

.....

.....

.....

[3 marks]

b) Describe with examples how decomposition could be used to help program this task.

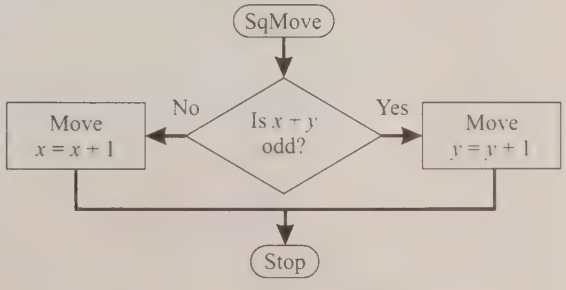
.....

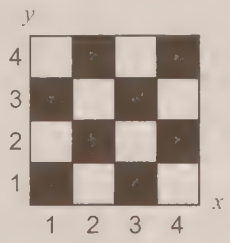
.....

.....

[3 marks]

4 A robot moves on the 4 × 4 square grid shown below.
A subroutine, SqMove, is part of a flow diagram that tells the robot how to move.





Draw a flow diagram to show how the robot moves. The flow diagram should:

- Ask the user to enter which square the robot starts on.
- Run the subroutine SqMove on a loop.
- Stop when the robot reaches the top or the right of the grid.

This is when x = 4 or y = 4

[6 marks]

Search Algorithms

Computers need to follow search algorithms to find items in a list — the ones you'll need to know about are binary search and linear search. Now, if only someone could make a search algorithm to find my keys.

A Binary Search looks for items in an Ordered List

BINARY SEARCH ALGORITHM

1) Find the middle item in the ordered list.
2) If this is the item you're looking for, then stop the search — you've found it.
3) If not, compare the item you're looking for to the middle item.
   If it comes before the middle item, get rid of the second half of the list.
   If it comes after the middle item, get rid of the first half of the list.
4) You'll be left with a list that is half the size of the original list.
   Repeat steps 1) – 3) on this smaller list to get an even smaller one.
   Keep going until you find the item you're looking for.

To find the middle item in a list of n items do $(n + 1) \div 2$ and round up if necessary.

EXAMPLE: **Use the binary search algorithm to find the number 99 in the following list.**

7	21	52	59	68	92	94	99	133

There are 9 items in the list so the middle item is the $(9 + 1) \div 2 = 5$th item.
The 5th item is 68 and $68 < 99$ so get rid of the first half of the list to leave:

92	94	99	133

There are 4 items left so the middle item is the $(4 + 1) \div 2 = 2.5 = 3$rd item
The 3rd item is 99. You've found the item you're looking for so the search is complete.

A Linear Search can be used on an Unordered List

A linear search checks each item of the list in turn to see if it's the correct one.
It stops when it either finds the item it's looking for, or has checked every item.

LINEAR SEARCH ALGORITHM

1) Look at the first item in the unordered list.
2) If this is the item you're looking for, then stop the search — you've found it.
3) If not, then look at the next item in the list.
4) Repeat steps 2) – 3) until you find the item that you're looking for or you've checked every item.

1) A linear search is much simpler than a binary search but not as efficient. A linear search can be used on any type of list, it doesn't have to be ordered. Due to it being inefficient, a linear search is often only used on small lists.
2) Once the list has been ordered, a binary search is much more efficient than a linear search. In general a binary search takes fewer steps to find the item you're looking for, which makes it more suitable for large lists of items.

EXAMPLE:

Use a linear search to find the number 99 from the list above.

Check the first item: $7 \neq 99$
Look at the next item: $21 \neq 99$
Look at the next item: $52 \neq 99$
Look at the next item: $59 \neq 99$
Look at the next item: $68 \neq 99$
Look at the next item: $92 \neq 99$
Look at the next item: $94 \neq 99$
Look at the next item: $99 = 99$

You've found the item you're looking for so the search is complete.

EXAM TIP

Write out every step of a search algorithm, don't skip ahead...

The binary and linear search algorithms might seem like a faff for you to follow when you can just look at a list and pick out the item you want. Sadly computers are more systematic and they need to follow every step of an algorithm — in your exam you'll need to show every step too.

Sorting Algorithms

I'm sure you all know how to sort things into numerical or alphabetical order but try telling a computer that. You'll need to be able to follow and carry out the three sorting algorithms on the next three pages.

A **Bubble Sort** compares **Pairs** of items

The bubble sort algorithm is used to sort an unordered list of items.
The algorithm is very simple to follow but can often take a while to actually sort a list.

BUBBLE SORT ALGORITHM

1) Look at the first two items in the list.
2) If they're in the right order, you don't have to do anything.
   If they're in the wrong order, swap them.
3) Move on to the next pair of items (the 2nd and 3rd entries) and repeat step 2).
4) Repeat step 3) until you get to the end of the list — this is called one pass.
   The last item will now be in the correct place, so don't include it in the next pass.
5) Repeat steps 1) – 4) until there are no swaps in a pass.

Each pass will have one less comparison than the one before it.

EXAMPLE: **Use the bubble sort algorithm to write these numbers in ascending order.**

66	21	38	15	89	49

First pass:

66	21	38	15	89	49	Compare 66 and 21 — swap them.
21	66	38	15	89	49	Compare 66 and 38 — swap them.
21	38	66	15	89	49	Compare 66 and 15 — swap them.
21	38	15	66	89	49	Compare 66 and 89 — no swap.
21	38	15	66	89	49	Compare 89 and 49 — swap them.
21	38	15	66	49	89	End of first pass.

After the 2nd pass the order of the numbers will be: 21 15 38 49 66 89
After the 3rd pass the order of the numbers will be: 15 21 38 49 66 89
There are no swaps in the 4th pass so the list has been sorted: 15 21 38 49 66 89

The bubble sort is considered to be one of the simplest sorting algorithms as it only ever focuses on two items rather than the whole list of items.

Pros

- It's a simple algorithm that can be easily implemented on a computer.
- It's an efficient way to check if a list is already in order. For a list of n items you only have to do one pass of $n - 1$ comparisons to check if the list is ordered or not.
- Doesn't use very much memory as all the sorting is done using the original list.

Cons

- It's an inefficient way to sort a list — for a list of n items, the worst case scenario would involve you doing $\frac{n(n-1)}{2}$ comparisons.
- Due to being inefficient, the bubble sort algorithm does not cope well with a very large list of items.

EXAM TIP

In the bubble sort, items bubble up to the end of the list...

A common mistake is to forget the final pass because you realise that the list is already in order, remember that you should always show a pass when nothing changes to complete the algorithm.

Sorting Algorithms

The next sorting algorithm you'll need to learn is the merge sort — it splits a list apart and then magically merges it back together in the correct order. I really hope you're ready to see something special.

A Merge Sort Splits the list apart then Merges it back together

The merge sort algorithm is an example of a divide-and-conquer algorithm and takes advantage of two facts:

- Small lists are easier to sort than large lists.
- It's easier to merge two ordered lists than two unordered lists.

MERGE SORT ALGORITHM

1) Split the list in half (the smaller lists are called sub-lists) — the second sub-list should start at the middle item (see p.56).
2) Keep repeating step 1) on each sub-list until all the lists only contain one item.
3) Merge pairs of sub-lists so that each sub-list has twice as many items. Each time you merge sub-lists, sort the items into the right order.
4) Repeat step 3) until you've merged all the sub-lists together.

EXAMPLE: **Use the merge sort algorithm to write these letters in alphabetical order.**

1) Split the original list of 8 items into two lists, the second list should start at the $(8 + 1) \div 2 = 4.5 =$ 5th item.
2) Carry on splitting the sub-lists until each list only has one item in it.
3) Merge and order sub-lists back together. E.g.

Compare F and A — move A to the new list.
Compare F and L — move F to the new list.
Compare P and L — move L to the new list.
P is the last item in the new list.

Note that merging is always performed on two ordered lists and is very simple to do.

4) Keep merging sub-lists until you only have one list.

You'll often be unable to split or merge lists evenly. For example, sometimes you'll have to merge a list containing two items with a list containing one item to make a list of three items.

Pros	Cons
• In general it's much more efficient and quicker than the bubble sort (p.57) and insertion sort algorithms (p.59) for large lists. • It has a very consistent running time regardless of how ordered the items in the original list are.	• It's slower than other algorithms for small lists. • Even if the list is already sorted it still goes through the whole splitting and merging process. • It uses more memory than the other sorting algorithms in order to create the separate lists.

Due to its efficiency the merge sort algorithm or variations of it are used in many programming languages such as Java™, Python and Perl as the primary sorting algorithm.

EXAM TIP

To do the merge sort, split things down before sorting them...

When doing a merge sort it's important that you show the splitting process and the merging process — if you only show the merging process then you've only shown half the algorithm.

Sorting Algorithms

Here it comes, the final sorting algorithm you'll need to know for your exams. It's called the insertion sort — have a look below to see how it's done.

An **Insertion Sort** orders the items as it goes

The insertion sort algorithm is the simplest sorting algorithm to understand — it just takes each item in turn and puts it in the right place using the first item in the list as a starting point.

INSERTION SORT ALGORITHM

1) Look at the second item in a list.
2) Compare it to all items before it (in this case just the first item) and insert the number into the right place.
3) Repeat step 2) for the third, fourth, fifth, etc. items until the last number in the list has been inserted into the correct place.

EXAMPLE: **Use the insertion sort algorithm to write these words in alphabetical order.**

Ball Stamp Post Tackle Scrum Kick Rugby

Ball	Stamp	Post	Tackle	Scrum	Kick	Rugby	No insertion.
Ball	Stamp	Post	Tackle	Scrum	Kick	Rugby	Insert post between ball and stamp.
Ball	Post	Stamp	Tackle	Scrum	Kick	Rugby	No insertion.
Ball	Post	Stamp	Tackle	Scrum	Kick	Rugby	Insert scrum between post and stamp.
Ball	Post	Scrum	Stamp	Tackle	Kick	Rugby	Insert kick between ball and post.
Ball	Kick	Post	Scrum	Stamp	Tackle	Rugby	Insert rugby between post and scrum.
Ball	Kick	Post	Rugby	Scrum	Stamp	Tackle	All items are sorted into the correct order so the sorting is complete.

Insertion sorts have many advantages over the other sorting algorithms:

- It's an intuitive way of sorting things and can be easily coded.
- It copes very well with small lists — for this reason, an insertion/merge hybrid sort is often used to take advantage of the strengths of each algorithm.
- All the sorting is done on the original list so, like the bubble sort, it doesn't require very much additional memory.
- It's very quick to add items to an already ordered list.
- It's also very quick at checking that a list is already sorted.

However, like the bubble sort, its main disadvantage is that it doesn't cope well with very large lists.
For a list containing n items:

- Best case scenario (when the list is already ordered) requires $n - 1$ comparisons.
- Worst case scenario requires $\frac{n(n-1)}{2}$ comparisons.

Take the items one by one and put them into the right position...

The insertion sort is pretty straightforward but don't fall into the trap of just taking the list and rewriting it in order, that's not how a computer would do it. You should always show every step of the algorithm.

Warm-Up and Worked Exam Questions

There are lots of searching and sorting algorithms for you to learn, so here are some questions to test your knowledge. Start off with the warm-up questions then have a go at the exam questions.

Warm-Up Questions

1) Here is a list of the ages of Julie's sons and daughters:

3	6	8	11	13	15	18

   a) Use a binary search to find "8" in the list.
   b) Use a linear search to find "11" in the list.

2) Describe how a bubble sort works.

3) Here are the names of the four pupils who attend chess club.

Chris	Beth	Dalia	Ahmed

   a) Show the steps of a bubble sort to put the names in ascending alphabetical order.
   b) Show the steps of a merge sort to put the names in descending alphabetical order.
   c) Show the steps of an insertion sort to put the names in ascending alphabetical order.

4) Which sorting algorithm is best to use on large lists of items?

Worked Exam Questions

1 Nicola has a list of numbers: 2, 3, 7, 5, 13, 11.

a) She says, "I can't use a binary search to find 13." Why is this the case?

A binary search only works on ordered data.

[1 mark]

b) Show the steps of a linear search to find 13 in the list above.

Check the first item: 2 ≠ 13.
Check the second item: 3 ≠ 13.
Check the third item: 7 ≠ 13.
Check the fourth item: 5 ≠ 13.
Check the fifth item: 13 = 13.
Stop searching as the item has be found.

The linear search is straightforward but make sure you show every single step of the algorithm to get full marks.

[2 marks]

2 Use the insertion sort algorithm to put these European cities into alphabetical order.

Underlining the item that you're looking at will help you keep track of where you're up to.

Riga	Paris	Oslo	Baku	Minsk
Paris	Riga	Oslo	Baku	Minsk
Oslo	Paris	Riga	Baku	Minsk
Baku	Oslo	Paris	Riga	Minsk
Baku	Minsk	Oslo	Paris	Riga

[4 marks]

Exam Questions

3 Six athletes compete in a long jump competition. Their best jumps are shown below. Show the stages of a bubble sort to list these jumps in order from shortest to longest.

5.32 m	5.50 m	5.39 m	6.50 m	6.28 m	6.14 m

[4 marks]

4 Sonia has a sorted list of ice cream flavours that she sells in her shop.

a) Show the stages of a binary search to find the word 'butterscotch' in the list below.

butterscotch	chocolate	mint	strawberry	vanilla

Always check that lists are ordered before doing a binary search.

[4 marks]

b) Give **one** advantage of using a binary search over a linear search.

.....

.....

[1 mark]

5 Kim has a hand of playing cards, all of the same suit: 3, 7, 6, 2, 5.

a) Arrange the cards in order from highest to lowest using:

i) An insertion sort.

[4 marks]

ii) A merge sort.

[4 marks]

b) Explain **one** benefit of using an insertion sort over a merge sort on a computer.

.....

.....

[2 marks]

Revision Questions for Section Four

Well that's algorithms all done and dusted. Or so you thought — just wait until you start Section 5.

- Try these questions and tick off each one when you get it right.
- When you've done all the questions for a topic and are completely happy with it, tick off the topic.

Computational Thinking (p.51) ☐

1) What is meant by: a) decomposition? b) abstraction? ☐
2) Why is using algorithmic thinking useful when solving a problem? ☐
3) Outline the decomposition, abstraction and algorithmic processes for choosing a film at the cinema. ☐

Pseudocode and Flow Diagrams (p.52-53) ☐

4) What is pseudocode? Give three features of well-written pseudocode. ☐
5) What are the benefits of writing algorithms in pseudocode rather than a programming language? ☐
6) Draw the five box types used on flow diagrams and say what each one is used for. ☐
7) What do sequences, selections and iterations look like on a flow diagram? ☐
8)* Draw a flow diagram to check if a new username is valid. Usernames should be at least 5 characters long and unique. If it's invalid, the algorithm should give the reason why and get the user to enter another username. ☐

Search Algorithms (p.56) ☐

9) What are the four steps of the binary search algorithm? ☐
10) What are the four steps of a linear search algorithm? ☐
11)* Here's a fascinating list of British towns and cities:

Ashington	Brecon	Chester	Dagenham	Morpeth	Usk	Watford

a) Use a binary search to find "Morpeth" in the list above.
b) Now do the same using a linear search. ☐
12) What are the benefits and drawbacks of using a linear search over a binary search? ☐

Sorting Algorithms (p.57-59) ☐

13) a) What are the five steps of the bubble sort algorithm?
b)* Use the bubble sort algorithm to sort these fruit into alphabetical order:

Orange	Banana	Apple	Peach	Grape	Lime

☐
14) What are the four steps of the merge sort algorithm? ☐
15) And the three steps of the insertion sort algorithm? ☐
16)* Here is a list of numbers:

8	7	5	1	3	6	4	2

a) Use the merge sort algorithm to sort this list into descending order.
b) Use the insertion sort algorithm to sort this list into ascending order. ☐
17) Outline the strengths and weaknesses of the following sorting algorithms:
a) bubble sort b) merge sort c) insertion sort ☐

*Answers on p.143

Programming Basics — Data Types

It's been a long wait of 62 pages, but finally we've got to the headline act, the programming section...

Everything we cover in this section will work slightly differently in different programming languages, but the principles are the same and that's what you need to learn for the exam.

```
In this section, examples of code will be given in
these boxes and will be written in pseudocode (p.52).
```

```
The output of the code will be shown in this box.
```

Programming languages have Five Main Data Types

1) Programming languages store data as different types. You need to learn the ones in this table...

Data type	Pseudocode	Characteristics	Examples
Integer	int	Whole numbers only.	0, 6, 10293, –999
Real (or float)	real	Numbers that have a decimal part.	0.15, –5.87, 100.0
Boolean	bool	Can only take one of two values, usually true or false.	True/False, 1/0, yes/no
Character	char	A single letter, number, symbol.	"A", "k", "5", "–", "$"
String	string	Used to represent text, it is a collection of characters.	"FsTmQ2", "$money$"

2) Each data type is allocated a different amount of memory.
3) Using the correct data types makes code more memory efficient, robust (hard to break) and predictable.

Programming languages can be weakly typed or strongly typed.

- Weakly typed languages will try to convert data types to avoid errors, however this can lead to unpredictable results.
- Strongly typed languages won't try to convert data types and so will produce more errors but more predictable results.

Data type	Typical amount of memory taken up
Integer	2 bytes or 4 bytes.
Real	4 bytes or 8 bytes.
Boolean	1 bit is needed but 1 byte is usually used.
Character	1 byte
String	1 byte for every character in the string.

Using the Correct Data Type for different Variables

You should be able to choose the best data type to use in different situations.

EXAMPLE: **Give the appropriate data type for each of the categories in this registration form.**

Initial of first name:	N
Surname:	Chapman
Age (in whole years):	27
Height (in metres):	1.64
Male or Female:	Female

Initial of first name should be stored as a character.

Surname should be stored as a string.

Age (in whole years) should be stored as an integer.

Height (in metres) should be stored as a real data type.

Male or Female could be stored as boolean.

Using the wrong data type can lead to unexpected results...

Using the correct data types is a fundamental part of programming — sometimes a piece of data could take different data types and you'll have to decide which is best based on the context of the question.

Programming Basics — Casting and Operators

A mixed bag coming up on this page. First up is casting, which allows you to convert from one data type to another. Then it's arithmetic operators — you'll already be familiar with most of them from maths.

Casting is used to change the Data Type

1) Languages have functions (p.81) that let you manually convert between data types — this is known as casting. This can be done using the int(), float(), bool() and str() commands.

`int("1")` ← Converts the string "1" to the integer 1.

`float(1)` ← Converts the integer 1 to the float 1.0.

The int() and float() functions will only work on numbers.

`bool(1)` ← Converts the integer 1 to the Boolean value True.

`str(True)` ← Converts the Boolean value True to the string "True".

2) It's important to realise that the integer 1, the real 1.0 and the strings "1" and "1.0" are all different.

3) You can also find the ASCII number (see p.110) of characters and vice versa using the ASC() and CHR() functions.

`ASC("b")` ← Converts the character "b" into its ASCII number 98.

`CHR(98)` ← Converts the ASCII number 98 into its equivalent character "b".

The Basic Arithmetic Operators are straightforward

1) The arithmetic operators take two values and perform a maths function on them.
2) Addition, subtraction, multiplication and division operators do what you'd expect.
3) The exponentiation operator is used to raise a number to a power.
4) The DIV operator returns the whole number part of a division and the MOD operator gives the remainder.

Dividing integers might behave oddly in some programming languages, e.g. 5 / 2 may give the answer 2 instead of 2.5...

...using DIV and MOD can avoid these issues.

Function	Typical Operator	Example	Result
Addition	+	5 + 5	10
Subtraction	–	3 – 10	–7
Multiplication	*	4 * 8	32
Division	/	42 / 6	7
Exponentiation	^ or **	2^3 (= 2^3)	8
Quotient	DIV	20 DIV 3	6
Remainder (modulus)	MOD or %	20 MOD 3	2

5) These operators work on integers and real data values or a combination of the two.
6) Computers follow the rule of BODMAS (Brackets, Other, Division, Multiplication, Addition & Subtraction) — so take care when using operators to make sure your code is actually doing what you want it to. E.g. 2 + 8 * 2 will give 18. To do the addition first, use brackets: (2 + 8) * 2 will give 20.

Arithmetic Operators only work on integers and reals...

Get your head around these basic things now and you'll have a better chance of understanding the trickier stuff later on. The DIV and MOD operators might seem strange but they're really useful.

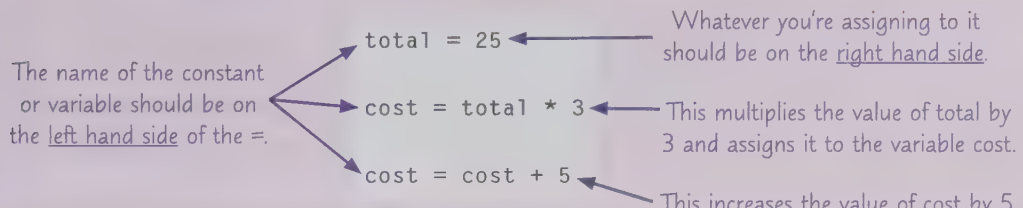
Programming Basics — Operators

Programming languages have other types of operator too — this page covers the assignment operator and the comparison operators.

The Assignment Operator

The assignment operator, =, is used to assign values to constants or variables (see next page).

Here are some examples of how the assignment operator works.



The Comparison Operators

Comparison operators compare the expression on their left hand side to the expression on their right hand side and produce a Boolean value (either true or false).

Comparison operator	What it means	Evaluates to True	Evaluates to False
==	Is equal to	5 == 5	5 == 8
<> or !=	Is not equal to	6 != 7	6 != 6
<	Is less than	4 < 10	3 < 2
>	Is greater than	15 > 9	10 > 12
<=	Is less than or equal to	7 <= 8	11 <= 10
>=	Is greater than or equal to	3 >= 3	9 >= 12

Don't get mixed up between = and ==...

1) A common mistake is to get the assignment operator = and the comparison operator == mixed up.
2) You'll know you've used them incorrectly because your code won't behave as intended.
3) Have a look at the examples below to see the impact using the wrong operator in an IF statement (p.70) can have.

```
if age = 25 then
```

This piece of code just assigns 25 to the variable "age" — the IF statement will consider this condition as always true.

```
if age == 25 then
```

This piece of code checks if age is equal to 25 and will only run if the condition is true.

Comparison operators are used to compare two expressions...

Knowing what all the different operators are and what they do is essential to learning how to write simple programs. Close the book and see if you can write down all the arithmetic and comparison operators, explain what each operator does and give an example of it being used.

Constants and Variables

Now that you know about the different data types and operations it's time to look at constants and variables. As you can probably tell by the names, constants remain the same and variables can be changed.

Data Values can be **Constants** or **Variables**

1) Data values can be stored as constants or variables.
2) The name of the constant or variable is linked to a memory location that stores the data value. The size of the memory location depends on the data type (see p.63).
3) A constant is assigned a data value at design time that can't be changed. If you attempt to change the value of a constant in a program then the interpreter or compiler (see p.92) will return an error.
4) Variables on the other hand can change value which makes them far more useful than constants.
5) Constants and variables often need to be declared (at the start of the program) before you can use them. This can be done in different ways:

'pressure' declared as a constant with integer type

'temperature' and 'weight' declared as variables with real type

```
const pressure as int = 30
var temperature as real = 20.5
var weight as real
```

Constants are assigned values when they're declared.

Initial values can be assigned to variables (but they don't have to be).

These lines of code do exactly the same as the lines above — temperature and weight are still variables.

```
const pressure = 30
real temperature = 20.5
real weight
```

To make code easier to follow, programmers usually follow standard naming conventions for constants and variables. E.g. 'lower case for the first letter, followed by a mixture of letters, numbers and underscores.'

Identifying **Constants** and **Variables** in Programs

EXAMPLE:

In a multi-event athletics competition, athletes get 5 points for winning an event and 2 points for coming second. Otherwise they get 0 points. This program calculates the total number of points that an athlete has.

```
firsts = input("Number of 1st places.")
seconds = input("Number of 2nd places.")
print(5 * firsts + 2 * seconds)
```

a) Rewrite the program so that all the variables are declared with data types and initial values.

The two variables are firsts and seconds. They should both be declared as integers as there are a whole number of events.

```
int firsts = 0
int seconds = 0
firsts = input("Number of 1st places.")
seconds = input("Number of 2nd places.")
print(5 * firsts + 2 * seconds)
```

The initial value of each variable is set to 0.

b) Give two reasons for assigning the values 5 and 2 to constants.

- They don't need to be changed during the running of the program.
- If the points awarded for each event was changed you'd only need to change the value given in the declaration of the constant.

This is an example of improving the maintainability (p.89) of the program.

A constant, a variable... and finally a constant. Time starts now...

You can't change the data type of a variable, only the value. But as you saw on p.64 you can use a casting function to return a different data type, which you can then assign to a new variable:

```
int cost = 50
string cost_string
cost_string = str(cost)
```

Strings

Remember from page 63 that strings are a data type made up of characters — these characters are alphanumeric (letters, numbers, spaces, symbols, etc.). Now you'll see how you can manipulate them.

Strings are written inside Quotation Marks

Strings are usually written inside double quotation marks ", but sometimes single quotes are used '.

```
string1 = "Print me, I'm a string."
print(string1)
```

```
Print me, I'm a string.
```

Strings can be joined together to form new strings — this is called concatenation. It's often done using the + operator.

```
string1 = "My favourite colour is"
string2 = "purple."
new_string = string1 + " " + string2
print(new_string)
```

The + operator joins the strings together

A space character has been added between the two strings.

```
My favourite colour is purple.
```

Programs let you Manipulate Strings in a variety of ways

1) Before getting started on string manipulation you should know that the characters in a string are usually numbered starting at 0.

2) Here are some common string manipulation functions that you'll need to learn for your exam.

Typical function	Operation	Effect on x = "Hello"
x.upper	Changes all characters in string x to upper case.	HELLO
x.lower	Changes all characters in string x to lower case.	hello
x.length	Returns the number of characters in string x.	5
x[i]	Extracts the character in position i from string x.	x[1] = "e"
x.subString(a, b)	Extracts a string starting at position a with length b from string x.	x.subString(3, 2) = "lo"

upper, lower, length and subString are special functions called methods. They act on a particular object (in this case strings) and are called using the object's name followed by a dot "." and the method's name.

EXAMPLE:

An electricity company generates a customer's 7 character username from:

- **the first 3 letters of their town as uppercase letters.**
- **the customer's age when they sign up (2 digits).**
- **the first and last letters of the customer's surname as lowercase letters.**

Write an algorithm to generate a username for any customer given that their data is stored under the variables town, age and surname.

Start by working out how to extract the information from each variable...

1) `town.subString(0, 3).upper` ← This extracts the first 3 characters from the customer's town and makes them uppercase.

2) `str(age)` ← Casts (p.64) the customer's age as a string.

3)
```
int n = surname.length
surname[0].lower + surname[n-1].lower
```
← Finds the length of the surname so that it can take the first and last characters and make them lowercase.

... then combine the code into a single algorithm at the end.

```
string username
int n = surname.length
username = town.subString(0, 3).upper + str(age) + surname[0].lower + surname[n-1].lower
```

EXAM TIP

I hope you don't just think I'm stringing you along...

It's important that you know the string manipulations on this page — the examiners might throw some different ones at you in the exam. Luckily they'll also show you exactly how they work.

Warm-Up and Worked Exam Questions

That's the first part of programming done — have a go at these questions to see how much you've understood.

Warm-Up Questions

1) What is the most appropriate data type for each of these items?
   a) The nickname of your best friend.
   b) The number on a rolled dice.
   c) The exact length of a car in metres.
   d) The answer to a yes/no question.

2) Work out the results of the following arithmetic operations:
   a) `5 * 8` b) `10 MOD 3` c) `28 DIV 6` d) `3 + 2 * 8`

3) Will the following pieces of code return true or false?
   a) `6 <= 10` b) `5 == 5` c) `14 > 15` d) `12 < 8 + 5`

4) What's the difference between a variable and a constant?

5) Given that `fish = "lobster"`, state what would be returned from the following methods:
   a) `fish.length`
   b) `fish.subString(0, 3)`
   c) `fish.upper`
   d) `fish.subString(2, 3)`

Worked Exam Questions

1 The program below calculates the cost of a burger in pounds at a fast food restaurant. A standard burger costs £6.50 with additional costs for toppings and eating in the restaurant.

```
const standard = 6.5
int toppings = 0
bool eat_in = false
toppings = input("How many toppings?")
eat_in = input("Are they eating in?")
if eat_in == true then
   print(standard + 0.5*toppings + 1)
else
   print(standard + 0.5*toppings)
endif
```

a) List **all** the variables in this program.

toppings and eat_in

[2 marks]

b) How much extra does it cost to eat your burger inside the restaurant?

£1

[1 mark]

c) The restaurant manager says that 0.5 should have been declared as a constant. Give **two** reasons for declaring this value as a constant.

1. It doesn't need to be changed as the program is running.
2. Updating the value of a constant once will update it everywhere in the program.

You could also mention that giving the value a name, e.g. toppingsCost will make the code more meaningful.

[2 marks]

2 A digital radio stores the date as a string under the variable name `date`. The radio is stuck on the date: `8 January 2016`. State the output from each of the following pieces of code:

a) `date.subString[0] + date.subString(10,4)`

"8" + "2016"

They are strings so use string concatenation instead of adding them.

82016

[1 mark]

b) `date.subString(2,3).upper`

JAN

[1 mark]

Exam Questions

3 A pedestrian crossing uses a button to request the traffic to stop. State the data type that you would use to record each of these variables and give a reason for your answer.

a) A variable to record whether the button has been pressed or not.

Data type: .....

Reason: .....

[2 marks]

b) A variable to record how many whole seconds it's been since the button was pressed.

Data type: .....

Reason: .....

[2 marks]

4 The program below calculates the value of an investment at the end of one year.

```
const investment
real rate
int interest
input investment
input rate
for x = 1 to 12
  interest = rate * investment
  investment = investment + interest
next x
print(investment)
```

Identify **two** problems with the constants or variables in the program.

1. .....

.....

2. .....

.....

[2 marks]

5 A juice company generates a product ID for each of its fruit juices. The product ID is generated using string concatenation on the first three letters of the fruit (in uppercase) and the volume of fruit juice in ml. E.g. a 500 ml carton of apple juice would be APP500.

a) Define what is meant by string concatenation.

.....

[1 mark]

b) What would the product code be for a 2000 ml carton of orange juice?

.....

[1 mark]

c) Complete the algorithm below so that line 05 reassigns the uppercase name of the fruit to the `fruit` variable and line 06 assigns the final product ID to the `prodID` variable.

```
01 string fruit, prodID
02 int volume
03 fruit = input("Enter the name of the fruit.")
04 volume = input("Enter the volume of the juice.")

05 fruit = .....

06 prodID = .....
07 print(prodID)
```

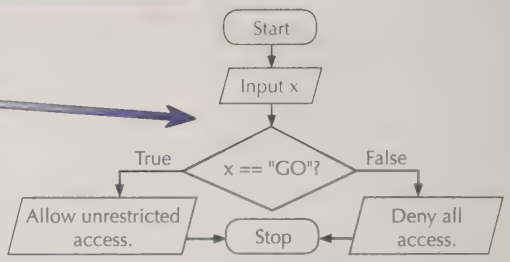
[3 marks]

Program Flow

The flow of a program is the order that the steps are carried out in. You can control the program flow using selection statements — there are two main types to learn, IF statements and SWITCH-CASE statements.

IF Statements usually have an if-then-else structure

1) IF statements allow you to check if a condition is true or false, and carry out different actions depending on the outcome. You can think about them as a flow diagram.



2) Here is a program that can verify if the user knows a certain passcode before granting access.

```
string x
x = input("Enter the passcode.")
if x == "GO" then
    Allow unrestricted access.
else
    Deny all access.
endif
```

The first part of the IF statement is the condition that must be checked.

Indenting the actions for each condition makes the code more readable.

The part after 'then' tells the program what to do if the condition is true.

The part after 'else' tells the program what to do if the condition is false.

3) If there is nothing for the program to do when the condition is false, leave out the 'else' part.

Nested IF Statements allow multiple outputs

1) More complex IF statements can be made by putting one IF statement inside another one — this type of selection statement is called a nested IF statement.

2) Nested IF statements allow you to check more conditions once you've established that the previous condition is true.

```
string x
x = input("Enter the passcode.")
if x == "GO" then
    if usertype == "Teacher" then
        Allow unrestricted access.
    else
        Allow restricted access.
    endif
else
    Deny all access.
endif
```

If the first condition is true, it will check the second condition...

If the second condition is true then unrestricted access is allowed.

If the first condition is false, it will run this else statement — all access is denied.

If the second condition is false then restricted access is allowed.

Indentation lets the reader see where each IF statement begins and ends.

3) IF-ELSEIF statements can also be used to check multiple conditions. They are different from nested IF statements as they only check more conditions if the previous condition is false.

```
if usertype == "Teacher" then
    Allow unrestricted access.
elseif usertype == "Parent" then
    Allow level 1 restricted access.
elseif usertype == "Pupil" then
    Allow level 2 restricted access.
else
    Deny all access.
endif
```

The conditions are all indented to the same level.

The first condition is always checked — if it's true then it will allow unrestricted access.

The second condition is checked if the first condition is false — if it's true then it will allow level 1 restricted access.

The third condition is checked if the first and second conditions are false — if it's true then it will allow level 2 restricted access.

If all conditions are false then the program will deny all access.

If you understand IF statements then make yourself a brew...

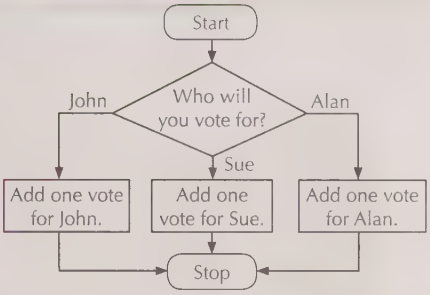
One good thing about IF-ELSEIF statements is they're very neat, everything is indented to the same line. Lots of nested IF statements with many levels of indentation can cause readability problems in your code.

Program Flow

Here are the SWITCH-CASE statements and the first type of iteration statement — FOR loops.

SWITCH-CASE Statements check the value of a Variable

1) Instead of checking to see if a statement is true or false, SWITCH-CASE statements can check if a variable has specific values.
2) They're used when you want a program to perform different actions for different values of the same variable.
3) Here is a program that can be used to count votes in an election.



The number of votes for each person starts at 0.

```
int johnvote = 0, suevote = 0, alanvote = 0
string vote
vote = input("Please cast your vote")
switch vote:
    case "John":
        johnvote = johnvote + 1
        print("You've voted for John.")
    case "Sue":
        suevote = suevote + 1
        print("You've voted for Sue.")
    case "Alan":
        alanvote = alanvote + 1
        print("You've voted for Alan.")
endswitch
```

The statement starts with 'switch vote' which uses the parameter 'vote' to decide which case to use.

If John is selected, add 1 to his vote and print "you've voted for John."

If Sue is selected, add 1 to her vote and print "you've voted for Sue."

Each 'case' should be indented to the same place.

If Alan is selected, add 1 to his vote and print "you've voted for Alan."

4) SWITCH-CASE statements have a similar structure to IF-ELSEIF statements but they give a neater way to test different values of a variable — this makes them easier than ELSEIF statements to maintain.
5) The drawback of SWITCH-CASE statements is that they can only check the value of one variable. IF-ELSEIF statements can check if multiple conditions are true.

FOR Loops are an example of a Count-Controlled Loop

1) FOR loops will repeat the code inside them a fixed number of times. The number of times that the code repeats will depend on an initial value, end value and the step count.
2) For example, for k = 1 to 10 step 3 will count up from 1 to 10 in steps of 3, so k = 1, k = 4, k = 7 and k = 10. If no step count is given the count will increase by 1 each time.

The FOR loop repeats the code between 'for' and 'next'.

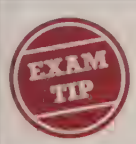
3) The number of times the loop repeats can also be set as the program runs — e.g. for k = 1 to x, where x is a variable.
4) FOR loops can also use the count within the loop — in the example on the right, k is used to keep track of how many votes have been cast.

Many programming languages don't use 'next' but it's used in pseudocode to make the code more readable.

```
int johnvote = 0, suevote = 0, alanvote = 0
string vote
for k = 1 to 100
    vote = input("Please cast your vote.")
    switch vote:
        case "John":
            johnvote = johnvote + 1
            print("You've voted for John.")
        case "Sue":
            suevote = suevote + 1
            print("You've voted for Sue.")
        case "Alan":
            alanvote = alanvote + 1
            print("You've voted for Alan.")
    endswitch
    print(str(k) + " votes have been cast.")
next k
```

Allows 100 votes to be cast.

The value of k can be used anywhere within the loop.



Use FOR loops when you know the number of repetitions...

In your exam a SWITCH-CASE statement might include a "default:" case. This tells the program what to do if none of the other cases are correct — it helps to make the statement more robust.

Program Flow

Here are a few other loops that you need to know about — the DO UNTIL loop, the WHILE loop and the DO WHILE loop. Like FOR loops, they are all iteration statements — they repeat a part of the program.

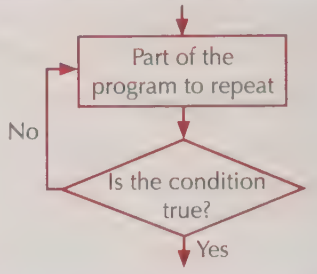
All these Loops are controlled by Conditions

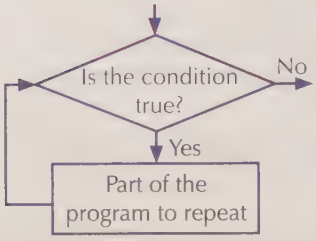
DO UNTIL, WHILE and DO WHILE loops are easy to get mixed up — they're very similar but with subtle differences that you need to know:

You might also see a REPEAT UNTIL loop — this does the same thing as a DO UNTIL loop

DO UNTIL LOOPS

- Controlled by a condition at the end of the loop.
- Keep going until the condition is true (i.e. while it is false).
- Always run the code inside them at least once.
- You get an infinite loop if the condition is never true.



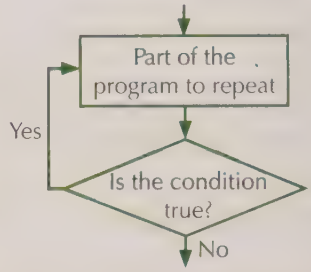


WHILE LOOPS

- Controlled by a condition at the start of the loop.
- Keep going while the condition is true (i.e. until it is false).
- Never run the code inside them if the condition is initially false.
- You get an infinite loop if the condition is always true.

DO WHILE LOOPS

- Controlled by a condition at the end of the loop.
- Keep going while the condition is true (i.e. until it is false).
- Always run the code inside them at least once.
- You get an infinite loop if the condition is always true.



EXAMPLE: **Write an algorithm that a supermarket self-scan machine could use to check if enough money has been fed into it and output the right amount of change.**

DO UNTIL Loop:

```
int total = 0
int cost, coin, change
cost = total cost in pence
do
 coin = input("Value of coin")
 total = total + coin
until total ≥ cost
change = total - cost
output change
```

The loop starts at 'do' and ends when the 'until' condition is true — when the total is greater than or equal to the cost.

WHILE Loop:

```
int total = 0
int cost, coin, change
cost = total cost in pence
while total < cost
 coin = input("Value of coin")
 total = total + coin
endwhile
change = total - cost
output change
```

The loop starts by checking the 'while' condition is true and keeps repeating until it is false — when the total is greater than or equal to the cost.

DO WHILE Loop:

```
int total = 0
int cost, coin, change
cost = total cost in pence
do
 coin = input("Value of coin")
 total = total + coin
while total < cost
change = total - cost
output change
```

The loop starts at 'do' and repeats until the 'while' condition is false — when the total is greater than or equal to the cost.

All of these loops in the example above work exactly the same when cost > 0.
If the cost is 0, the WHILE loop won't expect an input, whereas the DO UNTIL and DO WHILE loops will.

REVISION TIP

Keep looping through this page until it's stuck in your head...

As well as learning what each loop does, you should learn the difference between the different loops. The key thing is recognising exactly when the loop will start or stop.

Boolean Operators

Boolean operators work with Boolean values to produce a Boolean answer.

AND, OR and NOT are the only Boolean Operators you'll need

1) It doesn't make sense to use the arithmetic operators on things that are either true or false so instead you use the Boolean operators AND, OR and NOT.

Boolean operator	Examples that return true	Examples that return false
AND	3 < 5 AND 2 > 1	4 <= 5 AND 10 > 20
OR	1 > 8 OR 2 == 2	1 == 8 OR 2 < 2
NOT	NOT(5 > 8)	NOT(10 > 6)

In some code you might see AND written as &&, OR written as || and NOT written as !

2) Just like with numerical operators, you can combine Boolean operators — it's important that you use brackets in long Boolean expressions to let the computer know which part to do first. Boolean operations are carried out in the following order: brackets, NOT, AND then OR.

Boolean Operators can be used in Conditions

Boolean operators can be used to make all the different selection statements and iteration statements (p.70-72) more efficient and versatile.

EXAMPLES:

1. Karen and Stu are playing a 'best out of 10' game. The game should end when one of them wins 6 rounds or they both win 5 rounds. Write an algorithm to keep score in the game.

```
int karenrounds = 0, sturounds = 0
string roundwinner
do
    switch roundwinner:
        case "Karen":
            karenrounds = karenrounds + 1
        case "Stu":
            sturounds = sturounds + 1
    endswitch
until karenrounds == 6 OR sturounds == 6 OR (karenrounds == 5 AND sturounds == 5)
```

There is a SWITCH-CASE statement within the loop. This is where good indentation in your pseudocode is key.

The DO UNTIL loop stops when one of these three conditions is met.

2. In a computer game a character's status depends on three variables: hunger, hydration and comfort. If any of the conditions on the right are met then the character dies, otherwise they are alive.

- Any of the variables are equal to 0.
- Any two of the variables are less than 20.
- All three of the variables are less than 40.

Write an algorithm to work out the status of the character.

```
bool alive
if hunger == 0 OR hydration == 0 OR comfort == 0 then
    alive = false
elseif (hunger < 20 AND hydration < 20) OR (hunger < 20 AND comfort < 20)
       OR (hydration < 20 AND comfort < 20) then
    alive = false
elseif hunger < 40 AND hydration < 40 AND comfort < 40 then
    alive = false
else
    alive = true
endif
```

The Boolean operators are AND, OR and NOT...

Using Boolean operators can save lots of work by letting you check lots of conditions at the same time.

Warm-Up and Worked Exam Questions

Time to see if all that information is sinking in. Have a go at these warm-up questions then read through the worked exam questions before having a go at the questions on the next page yourself.

Warm-Up Questions

1) For each of the following, say if it is selection or iteration:
   a) DO UNTIL
   b) IF-THEN-ELSE
   c) SWITCH-CASE
   d) DO WHILE
   e) IF-ELSEIF
   f) WHILE

2) Give one difference between a SWITCH-CASE statement and an IF-ELSEIF statement.

3) a) Name a type of a count-controlled loop and a type of condition-controlled loop.
   b) Explain the difference between the two loops given in part a).

4) Decide whether the following Boolean expressions are true or false:
   a) 12 > 4 AND 8 == 5
   b) NOT(11 == 3)
   c) 12 <= 4 OR 10 != 5
   d) NOT(9 > 4 AND 5 < 2)

Worked Exam Questions

1 A garden centre has a climate monitoring system that gives warnings if the temperature and humidity aren't at suitable levels. The climate monitoring system contains this algorithm.

```
if humidity == 50 AND (temperature > 16 AND temperature < 25) then
   print("Humidity and temperature at acceptable levels.")
elseif temperature <= 16 OR temperature >= 25 then
   print("Warning – Please alter the temperature.")
else
   print("Warning – Please alter the humidity.")
endif
```

a) What will the output be if `humidity = 30` and `temperature = 16`?

"Warning — Please alter the temperature."

[1 mark]

b) What will the output be if `humidity = 30` and `temperature = 20`?

"Warning — Please alter the humidity."

[1 mark]

2 Salik needs a program that will ask users to create a password and then check if the password contains at least six characters. If it contains fewer than six characters the user must try again, otherwise the user is informed that their password is valid. Write an appropriate program for Salik.

```
string password
do
        password = input("Please enter a password")
until password.length >=6
print("Your password is valid")
```

Using a DO UNTIL or DO WHILE loop will mean that the code in the loop always runs through at least once.

[4 marks]

Exam Questions

3 Jasminda has written the following program to convert minutes into hours and minutes.

```
int minutes, hours, mins
minutes = input("Enter a number of minutes")
hours = minutes DIV 60
mins = minutes MOD 60
print(str(hours) + " hours and " + str(mins) + " minutes")
```

a) Is this an example of a sequence, selection or iteration? Tick the correct box.

Sequence ☐ Selection ☐ Iteration ☐

[1 mark]

b) What would the program print if the input was 150?

.....

[1 mark]

4 An electric heater has four temperature settings (0, 1, 2 and 3).
The code below controls the temperature of the heater.

```
int setting, temperature
switch setting:
    case 3:
        temperature = 50
    case 2:
        temperature = 30
    case 1:
        temperature = 20
    case 0:
        temperature = 0
endswitch
```

a) Rewrite this program using a different selection statement.

An ELSEIF statement will help you check lots of conditions.

[3 marks]

b) Give **two** reasons why a SWITCH-CASE statement is appropriate for this program.

1. .....

2. .....

[2 marks]

5 A tumble dryer will only be allowed to start if all of the following conditions are met:
- the real variable `weight` is more than 1.5 and less than 15.0
- the boolean variable `doorClosed` is true.

Write an algorithm that checks these conditions before allowing the tumble dryer to start.

[3 marks]

6 Karl and John are playing snap. Write an algorithm that:
- Asks for the name of the winner of each game.
- After ten games checks who has won more and displays the winner's name or tells them that it's a draw.

[6 marks]

Arrays

When you need to store data within a program you can do it using variables. But if you have lots of similar data to store, then using variables for each one is inefficient and that's where arrays come in.

Arrays are used to store multiple Data Values

1) An array is a data structure that can store a collection of data values all under one name.
2) Each piece of data in an array is called an element — each element can be accessed using its position (or index) in the array.
3) Arrays are most helpful when you have lots of related data that you want to store and it doesn't make sense to use separate variables — e.g. the names of pupils in a class, marks in a test, etc.
4) Just like variables, some languages require you to declare arrays before you use them.

A data structure is a format for storing data — other data structures include records, files and databases.

One-Dimensional Arrays are like Lists

The easiest way to get your head around one-dimensional arrays is to picture them as lists. Different languages have lots of fancy ways to create and update arrays. Here are the ones you'll need to learn for your exam:

1) Creating arrays — the first line of the code on the right creates the array 'rowers' and makes it size 4 (it can only contain 4 elements). The other lines assign the strings "Mark", "Adam", "Shelly" and "Tobias" in positions 0, 1, 2 and 3.

```
array rowers[4]
rowers[0] = "Mark"
rowers[1] = "Adam"
rowers[2] = "Shelly"
rowers[3] = "Tobias"
```

2) Retrieving elements from an array can be done by using the name of the array and the element's position. Remember that positions are numbered starting at 0.

```
print(rowers[0])
print(rowers[2])
```

```
Mark
Shelly
```

3) Changing elements is done by reassigning the array position to a different data value.

Replaces the rower in position 0 with "Tamal".

```
rowers[0] = "Tamal"
print(rowers)
```

```
["Tamal", "Adam", "Shelly", "Tobias"]
```

Notice that "Mark" has been completely removed from the array.

Combining these array functions with FOR loops (see p.71) will give you a systematic way of accessing and changing all of the elements in an array. Amongst other things, FOR loops can be used to search for specific elements, or make a similar change to lots of elements.

EXAMPLE: **The numbers below are stored in an array called scores[ ]. Write an algorithm that will add 3 to each number of the scores[ ] array.**

4	12	32	18	21	11	9	14	24

```
for k = 0 to 8
    scores[k] = scores[k] + 3
next k
```

For loop will run on each element of the array.

Adds 3 to the element in position k of the array.

REVISION TASK

Think of one dimensional arrays as lists of similar objects...

In some languages (e.g. C, C++, Java™) you'll find that arrays can only store one data type and that you can't change their size once they've been declared. For the programming language you're using, check if it uses arrays and look at what properties they have in that language.

Arrays

Now that you've covered one-dimensional arrays, the only way is up — that's right, two-dimensional arrays. Arrays can have even more dimensions, but luckily the examiners have decided that two is enough for now.

Two-Dimensional Arrays are like a List Of Lists

You can think of two-dimensional arrays as one-dimensional arrays where each element is also a one-dimensional array.

```
trees = [["oak", "ash"], ["beech", "cedar"], ["pine", "elm"]]
```

You can visualise arrays as tables or grids.

	0	1
0	oak	ash
1	beech	cedar
2	pine	elm

The position of an element is usually written as [a, b] or [a][b], where a represents the position of the one-dimensional list that the element is in and b represents its position within that one-dimensional list.

```
print("Ceara's favourite tree is" + trees[0, 0])
print("Shaun's favourite tree is" + trees[2, 1])
```

```
Ceara's favourite tree is oak
Shaun's favourite tree is elm
```

You can also change elements in exactly the same way as you saw for one-dimensional arrays (p.76).

EXAMPLE: **The 'scores' array has been used to store four test scores for five pupils, as shown. E.g. scores[2, 0] will return the test 2 score for pupil 0, which is 5.**

Tests \ Pupils	0	1	2	3	4
0	15	5	13	12	7
1	2	14	11	14	9
2	5	4	12	7	13
3	6	8	18	19	15

a) What will each of these return?

(i) scores[1, 3]

The entry in row 1 and column 3 is 14.

(ii) scores[3, 2] / scores[1, 0]

scores[3, 2] = 18 and scores[1, 0] = 2 so scores[3, 2] / scores[1, 0] = 18/2 = 9

b) Write an algorithm to count the total score of any given pupil.

As there aren't very many scores you could just add them together. E.g. for pupil 0 you could do scores[0, 0] + scores[1, 0] + scores [2, 0] + scores [3, 0]. But it's better practice to use a loop as it is easier to edit.

```
int total = 0
int pupil
pupil = input("Enter the number of the pupil.")
for i = 0 to 3
    total = total + scores[i, pupil]
next i
print total
```

c) The pass mark on every test was 9 or above. Write an algorithm to count the number of passes in the original array.

The passtotal variable keeps track of how many values are passes.

The i FOR loop searches each row and the j FOR loop searches each column.

The IF statement checks if the value in position [i, j] is a pass and adds 1 to passtotal if it is.

Finally passtotal is printed.

```
int passtotal = 0
for i = 0 to 3
    for j = 0 to 4
        if scores[i, j] >= 9 then
            passtotal = passtotal + 1
        endif
    next j
next i
print str(passtotal) + " marks were passes."
```

This is an example of a nested FOR loop.

Two-dimensional arrays are really just an array of arrays...

Two-dimensional arrays can be used to store information about a digital image — each pixel's information can be stored as an element in the array. Programmers can then manipulate the image using array commands, e.g. changing the values of pixels, cutting rows and columns out of the image, etc.

File Handling

File handling is all about how a program can access data and change data stored in an external file.

Always start by Opening the External File

This page will focus on how text is stored and accessed from ".txt" files. Other text files that are commonly used are ".csv" and ".dat".

1) Before you can do anything with a file you need to open it. This is done by using an open command and assigning it to a variable.

openRead will open the file in read mode — this allows you to read data from the file into your program.

This is the name of the file you want to open. Sometimes you'll have to give the whole file path.

```
newFile = openRead("newFile.txt")
newFile = openWrite("newFile.txt")
```

openWrite will open the file in write mode — this allows you to write data from your program to the file.

Some programming languages have other modes that allow you to do things like read and write at the same time.

2) Once a file is opened the program will start reading or writing from the beginning. As you read from or write to the file, the program keeps its place in the file (think of it like a cursor).

3) When you're finished reading or writing to a file you should always close it using close(). If you forget to close it then the file can remain locked and prevent others from editing it.

Read or Write to a file after it is Opened

1) After you have opened a file you can read or write to it depending on what mode it's in.

2) You can write lines of text to a file using writeLine(). If the file already contains some text then writeLine() will overwrite what is currently there.

The writeLine command is called on the variable that stores the external file.

```
winners = openWrite("victory.txt")
names = ["Jenny", "Carlos", "Matty", "Anna"]
for i = 0 to 3
    winners.writeLine(str(i) + " " + names[i])
next i
winners.close()
```

The writeLine command will automatically move onto the next line after it is called.

The text file will look like this.

```
0 Jenny
1 Carlos
2 Matty
3 Anna
```

3) You can read lines of text from a file using readLine().

Reads the first line of the text file as programs always start reading from the start of the file. After this command is called, the 'cursor' will be at the beginning of the second line.

```
winners = openRead("victory.txt")
first_line = winners.readLine()
second_line = winners.readLine()
winners.close()
```

Reads the second line of the file as that's where the program is up to. After this command is called, the 'cursor' will be at the beginning of the third line.

4) endOfFile() is another useful command that you'll have to know for your exams. It returns TRUE when the 'cursor' is at the end of the file. It's main use is to signify when a program should stop reading a file (like in this example).

```
array champions[]
int n = 0
winners = openRead("victory.txt")
while NOT winners.endOfFile()
    champions[n] = winners.readLine()
    n = n + 1
endwhile
print(champions)
```

```
["0 Jenny", "1 Carlos", "2 Matty", "3 Anna"]
```

Learning to read and write, it's like being back at primary school...

Data is stored externally so that it's not lost when the program is closed. E.g. a computer game will save your progress externally — if it was saved internally you'd lose your progress when the game was closed.

Storing Data

Records aren't just those big black round things that look like burnt CDs, they're also a useful data structure...

Records can contain Different Data Types

In the context of a database table, a record is just a row of data (see next page).

1) A record is a type of data structure (like an array — see p.76), which means that it is used to store a collection of data values.
2) One of the things that makes records so useful is that, unlike arrays, they can store values with different data types (see p.63), such as strings, integers and Booleans.
3) Each item in a record is called a field, and each field is given a data type and a field name when the record is created. The field name can help to describe the data stored in that field of the record.
4) Records are fixed in length, which means that you can't add extra fields to them once they've been created.

Different programming languages have slight variations on record data structures. E.g. Python has dictionaries, and C has structures.

Records can keep Related Information in one place

1) When you create a record structure, you can assign a data type and a name to each field:

```
record recipes
    int recipe_number
    string recipe_name
    bool tested
    int score
endrecord
```

The record is called 'recipes'.

'recipe_number', 'recipe_name', 'tested' and 'score' are the fields of the record.

Each field has its own data type.

2) Once you've created the structure of your record, you can assign it to variables:

'recipe1', 'recipe2' and 'recipe3' are all variables with the 'recipes' record structure.

```
recipe1 = recipes(1, "Chocolate Cake", True, 3)
recipe2 = recipes(2, "Lemon Slice", False, 0)
recipe3 = recipes(3, "Coconut Cookies", True, 8)
```

The data in each field needs to have the correct data type. E.g. the last one, 'score', should be an integer.

3) You can use the variable name to access a whole record. Alternatively, you can use the variable name along with a field name to access a particular item of a record.

```
print(recipe1)
print(recipe3.recipe_name)
```

```
(1, "Chocolate Cake", True, 3)
Coconut Cookies
```

Individual items in a record can be accessed and changed.

```
recipe2.tested = True
recipe2.score = 6
print(recipe2.recipe_name + " has a
      score of " + str(recipe2.score))
```

```
Lemon Slice has a score of 6
```

Arrays are handy if you want to Group Records together

If you have multiple variables with the same record structure, you can collect them in an array.

```
array recipeBook = [recipe1, recipe2, recipe3]
for i = 0 to 2
    if recipeBook[i].score >= 7 then
        print(recipeBook[i].recipe_name)
    endif
next i
```

```
Coconut Cookies
```

This will print the names of all recipes with a score ≥ 7.

You can visualise the recipeBook array as a table:

	recipe_number	recipe_name	tested	score
0	1	Chocolate Cake	True	3
1	2	Lemon Slice	True	6
2	3	Coconut Cookies	True	8

EXAM TIP

Well, we got through all that in record time...

You might see records presented differently to this, but the key concepts will be the same. You'll still need to understand what records and fields are, and how they are used in programming.

Searching Data

Structured Query Language (SQL) can be used to search tables (usually in a database) for specific data. When we talk about the records and fields of a database table we just mean the rows and columns.

SELECT and FROM are the most important keywords

In SQL, the SELECT keyword is followed by the names of the fields (columns) you to want to retrieve and display. Then the FROM keyword is followed by the name of the table (or tables) you want to search.

Table: hotels

ID	hotel_name	hotel_rating	rooms	bathroom	price_in_pounds
1	Water Lodge	2.3	50	En-suite	42
2	Fire Inn	4.2	64	Shared	42
3	Earthen House	4.4	215	En-suite	39
4	Windy Hotel	3.5	150	Shared	57
5	River Hotel	3.8	180	Shared	46

```
SELECT hotel_name
FROM hotels
```
This returns hotel_name for all the records in the table hotels.

```
SELECT hotel_name, hotel_rating
FROM hotels
```
This returns hotel_name and hotel_rating for all the records in the table hotels.

```
SELECT *
FROM hotels
```
If you want to return all the fields, you can use * as a wildcard.

You can use WHERE to filter the results

1) The WHERE statement is used to specify conditions that a record must satisfy before it is returned.

```
SELECT * FROM hotels WHERE hotel_rating >= 4.1
```
This condition looks for records with a hotel_rating greater than or equal to 4.1.

* is the wildcard character and will return all the fields.

Look in the table hotels.

ID	hotel_name	hotel_rating	rooms	bathroom	price_in_pounds
2	Fire Inn	4.2	64	Shared	42
3	Earthen House	4.4	215	En-suite	39

2) The boolean operators AND and OR can be used with WHERE to make more specific searches.

```
SELECT hotel_name FROM hotels WHERE bathroom = "En-suite" AND price_in_pounds < 45
```
Will only select the hotel_name.

Look in the table hotels.

This condition uses a Boolean operator to check if a hotel room has an en-suite bathroom AND is less than £45.

hotel_name
Water Lodge
Earthen House

3) The LIKE statement can also be used with WHERE to search for a pattern. In LIKE statements, the % character is used as a wildcard to represent any combination of letters and numbers.

```
SELECT hotel_name, price_in_pounds FROM hotels WHERE hotel_name LIKE "%Hotel"
```
Will only select hotel_name and price_in_pounds.

Look in the table hotels.

This will look for all hotel names ending with Hotel. The % character is used to show that it doesn't matter what comes before it.

hotel_name	price_in_pounds
Windy Hotel	57
River Hotel	46

EXAMPLE:

Stacey is searching for a hotel that has more than 100 rooms but she doesn't want to stay in the Windy Hotel. Write a search query for Stacey that returns the name and price of the hotels that match her requirements.

```
SELECT hotel_name, price_in_pounds FROM hotels
WHERE rooms > 100 AND hotel_name != "Windy Hotel"
```

hotel_name	price_in_pounds
River Hotel	46
Earthen House	39

This condition finds hotels with more than 100 rooms AND not called "Windy Hotel".

EXAM TIP

Tables of data can be searched using SQL...

There are lots of other SQL commands — in the exam you'll only have to know how to use the SELECT, FROM, WHERE and LIKE keywords and the wildcard characters * and %.

Sub Programs

Sub programs can be used to save time and to simplify code. By now you'll definitely have come across procedures and functions even if you don't know what they are yet — all is explained on the next two pages.

Procedures and Functions help to avoid Repeating Code

1) Procedures are sets of instructions stored under one name — when you want your program to do the whole set of instructions you only need to call the name of the procedure.
2) Functions are similar to procedures — the main difference is that functions always return a value.
3) Procedures and functions are very useful when you have sets of instructions that you need to repeat in different places within a program. They give your program more structure and readability whilst cutting down on the amount of code you actually need to write.
4) High-level programming languages (see p.92) have common procedures and functions built into them. If you want one that does something more specific you can create them yourself.
5) In most sub programs you'll encounter parameters and arguments so it's important that you know what they are and the difference between them:

- Parameters are special variables used to pass values into a sub program. For each parameter you can specify a name, a data type and a default value.
- Arguments are the actual values that the parameters take when the sub program is called.

Procedures carry out a Set Of Instructions

1) Procedures don't have to take parameters... ...but they sometimes will.

```
procedure welcome()
    print("Hello and welcome.")
    print("Let's learn about procedures.")
endprocedure
```

```
procedure betterwelcome(name)
    print("Hello " + name + " and welcome.")
    print("Let's learn about procedures.")
endprocedure
```

'name' is a parameter.

2) Procedures are called by typing their name (and giving an argument if necessary).

```
welcome()
```

```
Hello and welcome.
Let's learn about procedures.
```

```
betterwelcome("Pablo")
```

The betterwelcome procedure requires one argument.

```
Hello Pablo and welcome.
Let's learn about procedures.
```

3) Note that procedures don't return a value.

Functions will always Return a Value

1) Functions take at least one parameter and they must always return a value.
2) When a function is called it should be assigned to a variable or used in a statement otherwise the value that it returns will not be stored anywhere and will be lost.

EXAMPLE: **Write a function to join two strings together with a space between them and show it working on the strings "computer" and "science".**

See p.67 for a reminder on string manipulation.

```
function join_strings(x, y)
    return x + " " + y
endfunction
subject = join_strings("computer", "science")
print(subject)
```

A function should always return a value.

x and y are parameters.

"computer" and "science" are the arguments.

The result is stored as the variable 'subject'.

```
computer science
```

Functions will always return a value, procedures will not...

Even though they're similar, it's important that you don't get mixed up between functions and procedures.

Sub Programs

Make sure that you know your procedures from your functions and your parameters from your arguments.

Sub Programs can contain anything covered in this Section

EXAMPLE

Orla has been given a maths problem to add together all of the numbers between two integers (including the integers themselves) and work out if the total is divisible by 7. Write a function that Orla could use to solve the maths problem for any pair of integers.

The sub program is a function as it returns a value.

The variable 'total' is defined inside the function so it's a local variable (see below).

The IF statement checks if the total is divisible by 7. Remember % will give the remainder of a division (p.64).

```
function add_integers(x, y)
    int total = 0
    for i = x to y
        total = total + i
    next i
    if total % 7 == 0 then
        return true
    else
        return false
    endif
endfunction
```

x and y are the parameters of the function.

The FOR loop is used to add up all the integers from x to y.

Variables can be local or global

1) All variables have a scope (either local or global) — the scope of a variable tells you which parts of the program the variable can be used in.

All parameters have local scope to the sub program.

> Local variables can only be used within the structure they're declared in — they have a local scope.
> Global variables can be used any time after their declaration — they have a global scope.

2) Variables declared inside a sub program are local variables. They are invisible to the rest of the program — this means that they can't be used outside the function.

3) The advantage of local variables is that their scope only extends to the sub program they're declared in. They can't affect and are not affected by anything outside of the sub program. It also doesn't matter if you use the same variable name as a local variable defined elsewhere in the program.

4) Variables in the main body of a program can be made into global variables using the 'global' keyword — these variables can then be used anywhere in the program. It can be difficult to keep track of the value of global variables in larger programs.

5) The example below shows how global variables are used to store data outside of the sub program.

x and y are defined globally — if they were declared inside the sub program then they'd reset to 0 each time the sub program was called.

The sub program is a procedure as it doesn't return a value.

The parameters a and b are added to the global variables x and y.

The program keeps track of the position after the first move and then applies the second move from that position.

```
// A sub program to keep track of a character's x and y position.
global x = 0
global y = 0
procedure move(a, b)
    x = x + a
    y = y + b
    print("You're in square (" + str(x) + ", " + str(y) + ").")
endprocedure
move(3, 5)
move(4, 7)
```

a and b are parameters so they have local scope to this procedure — they're invisible elsewhere in the program.

```
You're in square (3, 5).
You're in square (7, 12).
```

It's considered good practice to use local variables wherever possible...

You should be able to identify the local and global variables in a program and use them in your own programs.

Warm-Up and Worked Exam Questions

That's all the learning done for the programming section — perfect time for some practice. Here are some warm-up questions to get you started, followed by a whole bunch of exam questions on the next two pages.

Warm-Up Questions

1) Explain what the following commands will do with the array `footballers`.
   a) `player = footballers[3]` b) `footballers[5] = "Pele"`
2) Write the pseudocode command that will:
   a) Open a file in read mode. b) Open a file in write mode.
   c) Close a file. d) Read a line of text from a file.
3) Give two reasons why a programmer may choose to use a record to store data.
4) Explain what each of these SQL keywords should be immediately followed by:
   a) SELECT b) FROM c) WHERE d) LIKE
5) Give three benefits of using sub programs when you are writing code.
6) What is the difference between arguments and parameters?

Worked Exam Question

1 Frances has written a list of jobs she has to do and stored it in the ToDoList.txt file shown on the right.

1. Clean my room.
2. Computer Science homework.
3. Organise my stamp collection.

a) Describe what each line of the code below does.

openRead(), readLine() and close() are three commands you'll need to be familiar with for your exams.

```
01 myList = openRead("ToDoList.txt")
02 print(myList.readLine())
03 myList.close()
```

Line 01 Opens the file in read mode and stores it under the variable myList.

Line 02 Prints the first line of the file, i.e. 1. Clean my room

Line 03 Closes the file.

[3 marks]

Frances writes the following code to add an extra job to the bottom of her list.

You'll also need to know what the openWrite() and writeLine() commands do.

```
myList = openWrite("ToDoList.txt")
myList.writeLine("4. Make lunch for parents.")
myList.close()
```

b) Explain why the code Frances has written will not work as intended.

The openWrite command will start writing from the start of the file so the first line of the text file will be overwritten with this text instead of it being added to the end.

[2 marks]

Exam Questions

2 Write a function that takes an integer as a parameter and returns the difference between the integer's cube and its square.

[3 marks]

3 A 2D array is used to store the names of the top 3 pupils in each event of a sports day.

a) What data type should each element of the array be assigned?

.....

[1 mark]

b) Give **three** reasons for using a 2D array to store this data.

1. .....

2. .....

3. .....

[3 marks]

4 The `cars` table below shows some data on the used cars that a car dealership has in stock:

CarID	Registration	Make	Type	Price	Engine_size
1	NF09 APY	Stanton	Hatchback	2500	1.4
2	SZ15 LUY	Fenwick	Saloon	4800	1.8
3	FQ55 ALW	Stanton	Hatchback	1700	2.1
4	SQ57 TTW	Fenwick	Estate	2300	2.8
5	NZ12 MBE	Stanton	Saloon	5200	1.8

a) How many records does this table have?

.....

[1 mark]

b) Explain the difference between a record and a field.

.....

.....

.....

[2 marks]

c) Draw a table showing what would be returned by the following SQL command:

```
SELECT Make, Type FROM cars WHERE Registration LIKE "N%"
```

[2 marks]

Exam Questions

5 John and three of his friends are training to run a marathon. John records how many miles he and three friends ran each day last week. John stores the data in a 2D array called `distanceRun`.

		Days of the week						
		0	**1**	**2**	**3**	**4**	**5**	**6**
Runner	**0**	9	10	8	12	0	6	9
	1	10	12	15	15	0	0	10
	2	15	14	13	16	0	8	9
	3	6	8	9	10	12	12	0

The distance run on day 0 by runner 2 is given by `distanceRun[0, 2]`.

a) Write the code to display the distance run on day 4 by runner 3.

.....

[1 mark]

b) Write an algorithm that takes a runner number as an input and outputs the total number of miles that they ran over the week.

[4 marks]

c) John has written the function `milesConvert()` which takes a distance in miles and returns the equivalent distance in km. E.g. `milesConvert(5)` would return 8. Write an algorithm to convert all distances in the array to km.

[3 marks]

6 Omar has written an adventure story in the file adventure.txt.
Write an algorithm that allows a user to print Omar's adventure story one line at a time.

- Each time the user presses the "y" key the next line of the story should be printed.
- The algorithm should end when it's at the end of the text file.

You'll need to use the endOfFile() command.

[5 marks]

7 The function `roll(n)` simulates the outcome of one random roll of an n-sided dice.
E.g. `roll(6)` will randomly return either 1, 2, 3, 4, 5 or 6.

a) Noel has declared a local variable inside the function.
Explain **two** differences between local variables and global variables.

1. .....

.....

2. .....

.....

[4 marks]

b) Noel wants to use his function in a dice game where two identical dice are rolled together.

- The player can choose the number of sides that the dice have.
- The player's score is the number of rolls it takes until both dice land on the same number.

Write a sub program that takes the number of sides of the dice as a parameter and returns a player's score.

[5 marks]

Revision Questions for Section Five

Well, that just about wraps up the programming section, perfect time to try some revision questions I think.

- Try these questions and tick off each one when you get it right.
- When you've done all the questions for a topic and are completely happy with it, tick off the topic.

Data Types, Operators, Constants, Variables and Strings (p.63-67) ☐

1) Define the following data types: integer, real, boolean, character and string. ☐

2) a) What does casting mean?
   b) Explain what each of these functions do:
   (i) int() (ii) float() (iii) str() (iv) ASC() (v) CHR() ☐

3) What do each of these operators mean?
   a) == b) != c) <= d) = e) ^ ☐

4) What is meant by: a) a constant? b) a variable? ☐

5) a) Define string concatenation and give an example of it being used.
   b) Explain what the following string manipulation methods do:
   (i) x.upper (ii) x.lower (iii) x.length (iv) x.subString(a, b) ☐

Program Flow and Boolean Operators (p.70-73) ☐

6) In 20 words or less, outline what each of these statements does:
   a) IF statement b) SWITCH-CASE statement. ☐

7) What is the main difference between IF-ELSEIF statements and nested IF statements? ☐

8) Compare the features of the three condition-controlled loops, DO UNTIL, WHILE and DO WHILE. ☐

9)* Write an algorithm that outputs the number of Mondays in a 30-day month when the user inputs the day of the week that the month started on. ☐

Arrays (p.76-77) ☐

10) Why are arrays useful? ☐

11)* Write commands to perform the following operations on this array. The name of the array is 'chars'.
   ["3", "T", "P", "2", "M", "e", "4", "q", "s", "3"].
   a) Print the character "M".
   b) Print the chars array with "P" changed to "D".
   c) Print the chars array with every element changed to "N". ☐

12)* Write an algorithm to create a two-dimensional array with 10 rows and 10 columns where each element is an integer and its value is given by the row number multiplied by the column number. (Hint: Remember that rows and columns are numbered starting at 0.) ☐

File Handling, Storing Data and Searching Data (p.78-80) ☐

13) Briefly describe what each of the following functions do:
   a) openRead() b) openWrite() c) endofFile() d) writeLine() e) readLine() ☐

14) In programming, data can be stored in records:
   a) What is a record? b) Give two differences between a record and an array. ☐

15)* Outline what the SQL query given below will do: ☐

```
SELECT * FROM world_records WHERE sport = "athletics" AND surname LIKE "M%"
```

Sub programs (p.81-82) ☐

16) What is the difference between a function and a procedure? ☐

17) Define these terms:
   a) parameter b) argument c) local variable d) global variable ☐

*Answers on p.145

Defensive Design

On p.32 you saw that insecure databases can be a security threat — unfortunately, every program that interacts with a user can be a risk. To keep programs safe from tampering you need to use defensive design.

Defensive Design helps to ensure programs Function Properly

1) When programs are functioning correctly they should never break and never produce errors. In practice this is difficult to achieve — even the biggest software companies need to update and patch their programs regularly.
2) Programmers try to protect their programs through defensive design. They will try to:

- Anticipate how users might misuse their program, then attempt to prevent it from happening.
- Ensure their code is well-maintained (see p.89).
- Reduce the number of errors in the code through testing (see p.90-91).

Misuse refers to the user doing things that you don't expect them to.

Make sure the Inputs can't be Exploited

1) The easiest way for a user to accidentally or intentionally misuse a program is when entering data.
2) There are two ways that you can prevent users from entering something you don't want them to:

INPUT SANITISATION

Removing any unwanted characters before passing data through the program.

INPUT VALIDATION

Checking if data meets certain criteria before passing it into the program. E.g. checking that an email address contains an @ symbol and has a suitable ending (.com, .co.uk, etc).

3) Here are a few types of input validation check you can use:

Range check	Checks the data is within a specified range.
Presence check	Checks the data has actually been entered.
Check digit (see p.100)	Checks numerical data has been entered accurately.
Format check	Checks the data has the correct format (e.g. a date).
Look-up table	Checks the data against a table of acceptable values.
Length check	Checks the data is the correct length.

4) Programs can use a mixture of input sanitisation and validation to verify that the inputted data has an acceptable format before passing it into the program.

The best defence is — well, a good defence...

Defensive design is something that you need to consider whenever you are writing a program. A well-designed program shouldn't just stop working if it experiences something that it doesn't expect — it should continue running and inform the user about what they've done wrong or how to correct it.

Defensive Design

It would be nice if input sanitisation and validation happened by magic, but I'm afraid that it usually requires some tricky coding to get right. Make sure you can still remember everything from Section Five...

You can **Sanitise** and **Validate Inputs** in your programs

EXAMPLE:

Karen is designing a program that she can use to create a database of file names. She has written the following function to check an inputted file name. The removeChar(x) method returns a new string with the character in position x removed.

```
function formatName(file)
    int x = 0
    while x < file.length
        switch file[x]:
            case "(":
                file = file.removeChar(x)
            case ")":
                file = file.removeChar(x)
            case default:
                x = x + 1
        endswitch
    endwhile
    return file
endfunction
```

a) **Explain what the function formatName() does.**

It goes through each character of a string and deletes the character if it's "(" or ")" and returns the amended string.

b) **Is this an example of input sanitisation or input validation?**

It gets rid of unwanted characters from the file name so this is an example of input sanitisation.

c) **Give two validation checks Karen could use to check that a file name has been entered and that it isn't too long.**

- Presence check to make sure data has been entered.
- Length check to make sure the data is not greater than the maximum length allowed.

Authentication can help **Protect** your programs

1) Authentication can confirm the identity of a user before they're allowed to access certain pieces of data or features of the program. A common way that programs do this is using passwords.
2) Passwords are usually associated with a username. When someone tries to access a protected part of the program, it should ask them for their password to check that they are who they claim to be.
3) Here are some common ways to increase the security of a password-based authentication system:

- Force users to use strong passwords (see p.33) and get them to change their passwords regularly.
- Limit the number of failed authentication attempts before access to an account is lost.
- Ask for a random selection of characters from the password on each authentication.

4) It's important that programmers get the level of authentication correct — too much authentication can affect a program's functionality and put people off using it.

Authentic revision materials, available to all users of this book...

When you're writing programs you should think about what you do and don't want a user to enter, but you can put too many restrictions on the data inputted to your program. When validation, sanitisation and authentication start to negatively impact the user's experience, then you've got too much defensive design.

Defensive Design

Programmers working on large projects might often have to work with code that somebody else wrote. Keeping your code tidy is like keeping your handwriting neat — it makes it easier for people to read it.

Good code should be **Well-Maintained**

1) As part of the defensive design of a program, programmers should make sure that it is well-maintained.
2) A well-maintained program makes it easy for other programmers to understand what the code does. They should also be able to change parts of the source code without the risk of causing problems elsewhere in the code (e.g. knock on effects).
3) The following features can improve the maintainability of source code:

- Comments (usually written after // or #) are useful for explaining what the key features of a program do — well written and clear comments are fundamental for helping other programmers understand your programs.
- Indentation can be used to separate different statements in a program. This allows other programmers to see the flow of the program more clearly and pick out the different features.
- Variables and sub programs should be named so that they refer to what they actually are. This helps programmers to keep track and recognise what the variables are all the way through your program.
- Only use global variables (see p.82) when necessary as they could affect the rest of your code. Variables with a local scope will only affect the sub programs that they are declared in — other programmers will know that changing these variables won't affect other parts of the program.

Too many comments can leave your programs looking cluttered and unreadable.

Well-maintained code is **Easy To Understand**

1) The example below shows some features of well-maintained code.

The sub program and variables have names which tell you what they are and what they do.

The code is indented so that you can see which bit of code falls within each statement.

Comments are used to tell the reader what the function does and what the FOR loop does.

```
// Converts a list of temperatures in °C to °F.
function convert_C_to_F(list_celsius)
    int list_length
    list_length = list_celsius.length
    array list_fahrenheit[list_length]
    // Converts each temperature in turn and adds them to a new list.
    for i = 0 to list_length - 1
        list_fahrenheit[i] = list_celsius[i] * 1.8 + 32
    next i
    return list_fahrenheit
endfunction
```

2) If a good amount of useful comments are put into the source code then it's very easy to produce a summary of what the program code actually does using auto-documentation (see p.93).

Keep your revision notes well-maintained, too...

Keeping code well-maintained isn't rocket science but sometimes programmers can get lazy and their code ends up in a bit of a mess. Have a look back at some code you've written. Write down some ways that you could have improved the maintainability of your code.

Testing

When you're writing programs, remember that the testing is just as important as the programming itself. Have a look at these pages to test your knowledge of testing — they'll prepare you for being tested in the tests.

Programming Errors can be **Syntax Errors** and **Logic Errors**

1) It's quite typical for a program to contain errors during its development — these errors need to be found and corrected as soon as possible.
2) The first task is to figure out what type of error has occurred:

> SYNTAX ERRORS — when the compiler or interpreter doesn't understand something you've typed because it doesn't follow the rules or grammar of the programming language.
> LOGIC ERRORS — when the compiler or interpreter is able to run the program, but the program does something unexpected.

3) Syntax errors can be diagnosed by compilers and interpreters (see p.92) — they'll be unable to turn the source code into machine code and a syntax error (with its location) will be returned.
4) Logic errors are more difficult to diagnose and track down — compilers and interpreters won't pick them up. Logic errors are found through general use of the program and by systematically testing it using a test plan (see p.91).

EXAMPLE

Jerry has written the following function. It multiplies a given positive integer by all the positive integers less than it (e.g. if the integer was 5 it should do 1 × 2 × 3 × 4 × 5). Identify two logic errors in Jerry's function and suggest how he should fix them.

```
function multiplier(n)
    for i = 1 to n
        int count = 1
        count = count * n
    next i
    return count
endfunction
```

Error 1: In line 3 the count variable is declared (and set to 1) within the loop, so each time the loop repeats, the count value will be set back to 1. The declaration of the count variable should be moved before the loop.

Error 2: In line 4 the count variable is multiplied by n, whereas it should be multiplied by i. It should read count = count * i.

Programs should be **Tested** before being released

1) Functionality testing is an essential part of the development process and a good way to spot logic errors.
2) The main aim of this testing is to see if the program actually meets your initial requirements, i.e. it does exactly what you wanted it to do without breaking or producing errors.
3) Functionality testing shouldn't be left until the end of the process — it's much better to spot errors and fix them as early as possible during development of the program.
4) Other types of testing will depend on what the program or system will be used for. For example:

Performance Test	Tests how quickly certain features run and their impact on computer resources.
Usability Test	Tests how user-friendly the interface and features are.
Security Test	Tests vulnerability to attacks and how securely data is stored.
Load/Stress Test	Tests how it copes under extreme conditions, e.g. lots of users at the same time.

There are more testing times ahead...

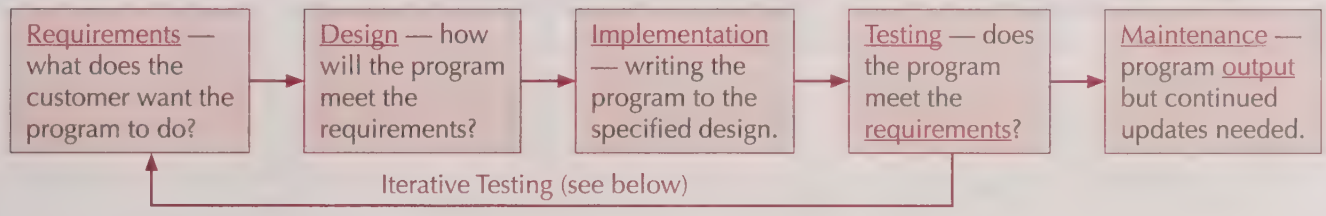
Syntax errors are usually easy to fix as the compiler should point you to the exact line that contains the error. On the other hand, it's often difficult for computers to help you out with logic errors — as far as they're concerned, if the program is running, everything is hunky-dory and working as it should.

Testing

So now that you know why programs are tested and what you're actually looking for, it's time to have a closer look at how they're tested. Often testing is planned out before development is even begun.

Testing is a key part of the Software Development Cycle

1) When developing a piece of software or a program you would traditionally follow this flow diagram.



2) The type of testing will impact the software development cycle.

- Final testing — the program only goes through the development cycle once. All the required features of the program are added at the same time. The program is tested against the initial requirements of the customer — if it meets them then the program is signed-off. The customer will get what they asked for but it won't necessarily be what they really want.
- Iterative testing — the program will go through the development cycle a few times. The idea is to try and get the program to match what the customer really wants. The requirements in the first cycle might only include the main features of the program. At the start of each new cycle the requirements will be adjusted (e.g. by adding new details and features).

A Test Plan should be made Before Implementation

1) A test plan will outline exactly what you're going to test and how you're going to test it. It should cover all the possible paths through a program and anticipate potential issues.

2) A good test plan will select appropriate test data to test for these issues. The test data that you use should fall into one of three categories:

Possible paths are all the branches of the flow diagram (p.53) for your program.

- Normal data — things that a user is likely to input into the program.
- Extreme (boundary) data — values at the limit of what the program should be able to handle.
- Erroneous data — inputs that the program should not accept.

3) The table below shows an example of a test plan for setting an alarm system. Users should be able to set their own 3-5 digit alarm code.

Type of data	Test data	Reason for testing	Expected outcome
Normal	2476	To see how the alarm copes with normal usage.	Code accepted.
Normal	No input	To see if the alarm prompts an input.	Prompt to enter a code.
Extreme	000	To see if the smallest code is accepted.	Code accepted.
Extreme	99999	To see if the largest code is accepted.	Code accepted.
Erroneous	23aY	To see if the system accepts non-digits.	Error: The code contains non-numeric data.
Erroneous	12	To see if the alarm accepts inputs shorter than 3 digits.	Error: The code is too short.
Erroneous	632191	To see if the alarm accepts inputs longer than 5 digits.	Error: The code is too long

4) During testing the tester can add "actual outcome" and "pass or fail" columns to the table.

Luckily for you, this page flew through its testing stage...

In the exam, you'll need to be able to design suitable testing procedures for a given scenario. This could include coming up with your own test plan and test data.

Translators

For computers to process any computer language it needs to be translated into machine code.

Computer Languages can be **High-Level** or **Low-Level**

1) Most of the programming languages that you'll be familiar with (e.g. Python, C++, Java™) are high-level languages. The source code is easy for humans to write, but computers need to translate it into machine code before they can read and run it.
2) On the other hand, low-level languages are tricky for humans to read and write but are easier for a computer to run. They consist of machine code and assembly languages:

```
000000 00010 00011 00100 00000 100000
```
← Binary machine code is very tricky for humans to understand.

```
ADD r4, r2, r3
```
← Assembly code is more readable for humans and easier to remember — the first bit (ADD) is the operation code and the rest tells you what to perform that operation on.

3) High-level languages are popular with programmers, but low-level languages have their uses too:

High-Level Languages	Low-Level Languages
• One instruction of high-level code represents many instructions of machine code. • The same code will work for many different machines and processors. • The programmer can easily store data in lots of different structures (e.g. lists and arrays) without knowing about the memory structure. • Code is easy to read, understand and modify. • Must be translated into machine code before a computer is able to understand it. • You don't have much control over what the CPU actually does so programs will be less memory efficient and slower.	• One instruction of assembly code usually only represents one instruction of machine code. • Usually written for one type of machine or processor and won't work on any others. • The programmer needs to know about the internal structure of the CPU (see p.2-3) and how it manages the memory. • Code is very difficult to read, understand and modify. • Commands in machine code can be executed directly without the need for a translator. • You control exactly what the CPU does and how it uses memory so programs will be more memory efficient and faster.

Translators convert programming languages into **Machine Code**

1) Computers only understand instructions given to them as machine code, so high level languages and assembly languages need to be translated before a computer is able to execute the instructions.
2) There are three types of translator that you need to know about: assemblers, compilers and interpreters.
3) Assemblers are used to turn assembly language into machine code. There are many different assembly languages (to support different CPU types) and each one needs its own unique assembler.
4) Compilers and interpreters are both used to turn high-level code into machine code.

Compiler	Interpreter
Translates all of the source code at the same time and creates one executable file.	Translates and runs the source code one instruction at a time, but doesn't create an executable file.
Only needed once to create the executable file.	Needed every time you want to run the program.
Returns a list of errors for the entire program once compiling is complete.	The interpreter will return the first error it finds and then stop — this is useful for debugging.
Once compiled the program runs quickly, but compiling can take a long time.	Programs will run more slowly because the code is being translated as the program is running.

5) The type of translator used will depend on which programming language and IDE (p.93) you're using.
6) If the program is stored over multiple source code files then a linker is used to join all of the separate compiled codes into one executable program.

"Cette page est incroyable!" — call in the translators...

You should know the key features of low- and high-level languages, assemblers, compilers and interpreters.

Integrated Development Environments

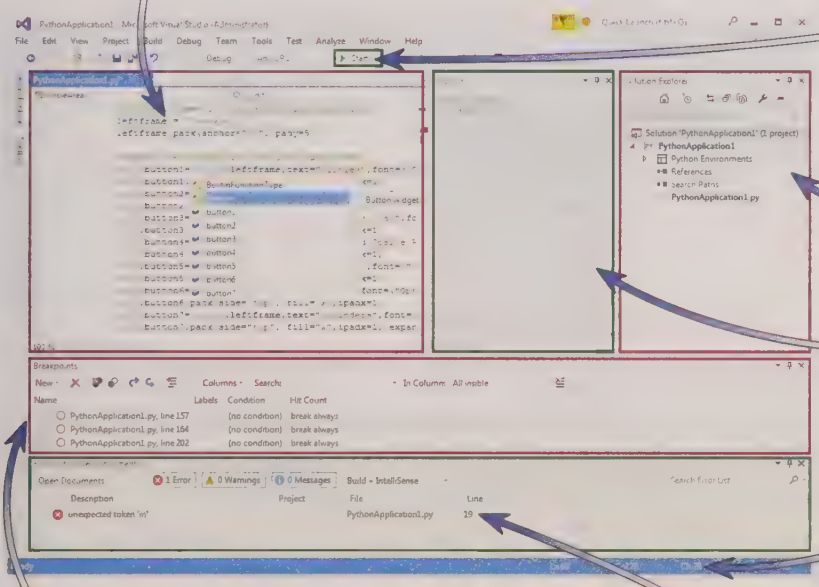
Integrated development environments (IDEs) provide programmers with lots of handy tools when they're coding. If there's one thing I know about programmers, it's that they'll do anything to make life a bit easier.

IDEs have lots of Features to help Programmers

An integrated development environment is a piece of software that provides features to help a programmer to develop their program. Most IDEs will have similar features — the example below shows some of the features from the Microsoft® Visual Studio® IDE:

The code editor is the main part of an IDE, it's where the code is written. Most code editors will have line numbering and auto-colour coding for things like strings, functions, variables and comments. Good code editors will also have other automatic features like auto-correct, auto-indentation and auto-complete.

A run-time environment allows the code to be run quickly within the IDE — this is done using a start or run button. The run-time environment can also help to identify logic errors in the program as the programmer can see which part of the code is running when errors occur.



An explorer window will help navigate through programs.

An output window to show the output from a program when it is run.

Features to keep track of the line number and character number that you're working on.

Breakpoints are a common debugging tool, they stop the program on certain lines so you can gather information like the value of variables as the program is running.

Error diagnostics and debugging tools help to find and fix errors (see p.90) in a program — they'll tell you the location of the error and often suggest ways to fix it.

Here are some other common features of an IDE:

- A translator (compiler, interpreter or both) which will translate the source code into machine code (see p.92). If the IDE has both then you can take advantage of each translator's best features.
- Auto-documentation helps with the maintenance of programs. It can extract certain features of a program, like the names of variables, names of sub programs and comments. This information is stored in a separate document to give a summary of what the code does.
- A Graphical User Interface (GUI) builder helps the programmer design a user interface by building it up graphically rather than having to design it using source code. It allows you to drag and drop different objects and customise them, so you can make the interface look exactly how you want it to.

IDEs can make developing programs much quicker and easier...

Each IDE has advantages and disadvantages so it's all about choosing one that meets your needs. For example, certain IDEs will have features that support multiple programming languages and others will specialise in one programming language — both types of IDE can be helpful in different situations.

Warm-Up and Worked Exam Questions

A whole section in one go — that's no mean feat. Make sure all of it has sunk in by having a go at these warm-up questions. Then, once you're happy, try the exam questions for some serious practice.

Warm-Up Questions

1) What is the difference between input validation and input sanitisation?
2) Name the type of input check that:
   a) checks that data has been entered. b) checks data against a table of possible values.
3) Give three features that will improve the maintainability of code.
4) Are the following statements true or false?
   a) Syntax errors are harder to find than logic errors.
   b) Using the wrong boolean operator is a logic error.
   c) Syntax errors will prevent code from running.
   d) Logic errors will prevent code from running.
5) Describe what is meant by 'iterative testing'.
6) Explain the purpose of a compiler.
7) Below is a sketch of the layout of a new Integrated Development Environment (IDE). In each box, briefly describe the purpose of that feature.

Code Editor	Breakpoints
Error Diagnostics	

Worked Exam Question

1 A retailer keeps a database of its loyalty card holders. The retailer stores the following data for each card holder: name, age, postcode and customer number.

Name	Age	Postcode	Customer No.
Carol Foreman	20	NE85 3TW	100278
Peter Taylor	55	HA55 8PZ	223327

a) Define the term input validation.

Checking data meets certain criteria before passing it through a program.

[1 mark]

b) Describe **two** suitable types of validation for an entry in the age field.

1. Format check to make sure the input is an integer.
2. Range check to make sure the age is sensible, e.g. between 0 and 120.

[2 marks]

c) Describe **two** suitable types of validation for an entry in the postcode field.

1. Length check to make sure the input has a valid number of characters.
2. Format check to make sure the input contains only letters and numbers.

A format check could also do things like check that the postcode ends with a number and two letters.

[2 marks]

Exam Questions

2 Cynthia is writing code for a tablet computer application aimed at children.
Explain how each of the following tools in the Integrated Development Environment (IDE) could help Cynthia write her application.

Translator: .....

.....

.....

Error Diagnostics: .....

.....

.....

Code editor: .....

.....

.....

[6 marks]

3 A holiday company has written a simple program to calculate the price of its group holiday packages. The program asks the user to input the group size — if the group size is smaller than two or greater than 10 the program displays an error message. If not, the price (in £s) is calculated by multiplying the group size by 50 and then adding 10.

a) Describe how the company can use a test plan to check for logic errors in the program.

.....

.....

.....

[3 marks]

b) Complete the test plan below by filling in the missing spaces.

Test Data	Expected Outcome	Reasons for test
Group_Size = 4		
	460	
Group_Size = 12		Check what happens if input too large.

[5 marks]

Exam Questions

4 A company specialises in writing programs using low-level languages.

a) Identify **two** reasons why some programmers might use low-level languages.

1. .....

2. .....

[2 marks]

b) Explain why programmers might prefer to use an assembly language over machine code.

.....

.....

[2 marks]

c) State the type of translator used to translate assembly languages into machine code.

.....

[1 mark]

5 Tiffany writes some code to check if an entered pincode is between 4 and 6 characters long.

```
string pincode = input("Enter pincode")
if pincode.length >= 4 OR pincode.length <= 6 then
    print("Valid pincode"
else
    print("Not a valid pincode, please try again")
endif
```

a) Identify the syntax error in Tiffany's code and suggest how she could correct it.

Error: .....

Correction: .....

[2 marks]

b) Identify the logic error in Tiffany's code and suggest how she could correct it.

Error: .....

Correction: .....

[2 marks]

6 A website's payment form requires users to input their credit card details, e.g. name, card number, expiration date, security code, etc. Evaluate the impact of only using input validation to check the details.

Think about problems that input validation can prevent, and those it can't.

[6 marks]

7 An exam board is developing automated software to calculate students' final exam grades. Evaluate the extent to which the different features of an IDE can help the exam board to develop a well-maintained program.

[6 marks]

Revision Questions for Section Six

And just like that it's the end of section six — but before you put the book down you've got one more task.

- Try these questions and tick off each one when you get it right.
- When you've done all the questions for a topic and are completely happy with it, tick off the topic.

Defensive Design (p.87-89) ☐

1) Why is it important for your programs to have a defensive design? ☐
2) Define the terms input sanitisation and input validation. ☐
3) Give six types of input validation check and explain what each check does. ☐
4)* The program below checks which year the user was born in.
Does this program use input sanitisation or input validation? Explain your answer.

```
int year
do
    year = input("Enter the year you were born.")
until year > 1900 AND year <= 2016
```

☐

5) What is authentication and why is it used? ☐
6) Give three things that can be done to make a password-based authentication system more secure. ☐
7) a) Give four features of maintainable source code.
b) Explain how each feature can help other programmers to maintain your code. ☐

Testing (p.90-91) ☐

8) Define the following terms: a) Syntax Error b) Logic Error ☐
9) Explain why logic errors are more difficult to diagnose than syntax errors. ☐
10)* This algorithm should take the user's age and always print one of the two strings.
Find two errors in the code and suggest ways that you could fix them.

```
int x = input("Enter your age")
if x > 16 then
    print("At the age of " + x + " you must be a computer science genius.")
elseif x < 16 then
    print("Practice makes perfect!")
endif
```

☐

11) Why is testing important? ☐
12) Give five types of testing that you might carry out on a program. ☐
13) What is meant by: a) iterative testing? b) final testing? ☐
14) What are the three different types of test data? ☐
15)* A software company is designing an anagram application. It will take a string of letters and return all of the words that can be spelt using all of the letters exactly once.
Come up with five pieces of test data that the company could use to test their program. ☐

Translators and IDEs (p.92-93) ☐

16) Define and give an example of the following: a) Machine code b) Assembly Language ☐
17) Give six differences between high-level languages and low-level languages. ☐
18) What are the three types of translator? ☐
19) Compare the functionality and uses of a compiler and an interpreter. ☐
20) IDEs have lots of different features. Explain what each of these features are:
a) Code Editor b) Run-time environment c) Error diagnostics
d) GUI builder e) Auto-documentation f) Breakpoints ☐

*Answers on p.146

Logic

Logic gates are pretty clever stuff. They take binary information and give an output based on the Boolean operations (p.73). Each Boolean operator (NOT, AND and OR) has its own logic gate.

Logic Gates apply Boolean Operations to Inputs

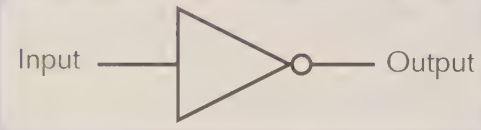
1) Logic gates are special circuits built into computer chips.
They receive binary data, apply a Boolean operation, then output a binary result.
2) Logic diagrams are often drawn to show logic gates and circuits.
Each type of logic gate is shown by a different symbol.
3) Each type of logic gate also has a corresponding truth table.
Truth tables show all possible input combinations of 1s and 0s, and the corresponding outputs.

NOT gate

1) NOT gates take a single input and give a single output.
2) The output is always the opposite value to the input.
If 1 is input, it outputs 0. If 0 is input, it outputs 1.

It can help to think of 1s as TRUE and 0s as FALSE.

NOT gate symbol



NOT truth table

Input	Output
0	1
1	0

AND gate

1) AND gates take two inputs and give one output.
2) If both inputs are 1, the output is 1, otherwise the output is 0.

AND gate symbol

AND truth table

Input A	Input B	Output
0	0	0
0	1	0
1	0	0
1	1	1

OR gate

1) OR gates take two inputs and give one output.
2) If one or more inputs are 1, then the output is 1, otherwise the output is 0.

OR gate symbol

OR truth table

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1

REVISION TIP

Logic isn't as scary as it looks...

These basic logic gates are the building blocks for bigger logic circuits. You should be able to draw each logic gate and the corresponding truth table — you'll also need to learn the equivalent expression and notation from this table:

Gate	Expression	Notation
NOT	NOT A	¬A
AND	A AND B	A ∧ B
OR	A OR B	A ∨ B

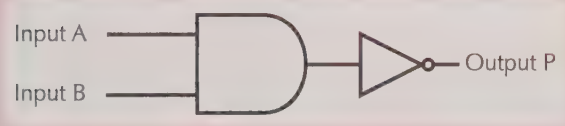
Logic

You can make more interesting logic diagrams by combining logic gates. If you know the truth tables from the previous page you'll be able to create truth tables for much more complicated logic diagrams.

Logic Gates are Combined for More Complex Operations

1) Multiple logic gates can be added to the same logic circuit to carry out different operations.
2) You can work out the truth tables by working through each gate in order.
For every input combination, follow them through each gate step-by-step, then write down the output.
3) By using brackets and the terms AND, OR and NOT, circuits can be written as logical statements, like NOT(A AND B) below. Operations in brackets should be completed first, just like in normal maths.

This circuit shows AND followed by NOT.



The truth table looks like this:

A	B	A AND B	P = NOT(A AND B)
0	0	**0**	**1**
0	1	**0**	**1**
1	0	**0**	**1**
1	1	**1**	**0**

This circuit shows OR followed by NOT.

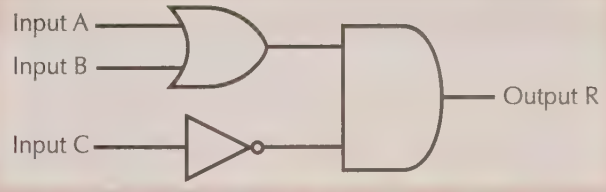
The truth table looks like this:

C	D	C OR D	Q = NOT(C OR D)
0	0	**0**	**1**
0	1	**1**	**0**
1	0	**1**	**0**
1	1	**1**	**0**

4) The two logic circuits shown above are examples of two-level logic circuits — they require the inputs to pass through a maximum of two logic gates to reach the output.

Logic Circuits can have More than Two Inputs

This is a two-level logic circuit with 3 inputs.



Using Boolean operators, this circuit can be written as R = (A OR B) AND (NOT C). (This is an example of Boolean algebra).

To cover every input combination, extra rows are needed in the truth table. There are 3 inputs and each can take one of 2 values, so $2 \times 2 \times 2 = 8$ rows are needed.

In general, the number of rows is 2^n, where n is the number of different inputs.

A	B	C	A OR B	NOT C	R = (A OR B) AND (NOT C)
0	0	0	0	1	**0**
0	0	1	0	0	**0**
0	1	0	1	1	**1**
0	1	1	1	0	**0**
1	0	0	1	1	**1**
1	0	1	1	0	**0**
1	1	0	1	1	**1**
1	1	1	1	0	**0**

To be OR NOT to be — literally covering all forms of being...

Once you've learned each gate's truth table, you can work out the truth tables of much more complicated circuits. If you take the inputs through each gate one step at a time you'll be fine — it's only logical...

Units

Just like you have units like centimetres, metres and kilometres for measuring distance, computers need units for measuring digital information. You'll need to learn all of the unit names on this page and their sizes.

Bits are the Smallest Measure of Data

1) Computers use 1s and 0s to represent the flow of electricity.
1 is used to show that electricity is flowing, and 0 shows that it is not flowing.
2) All the data we want a computer to process must be converted into binary code (1s and 0s).
3) Each 1 or 0 in a binary code is a bit (binary digit). For example, 1010 is 4 bits.
4) The table below shows the size of other units of data:

A byte is big enough to store one character (like x, e, M or £).
See p.110 for more info.

Most files (like songs, pictures and documents) are measured in kB or MB.

High definition videos and complex applications are often measured in gigabytes.

Secondary storage capacity is measured in gigabytes or terabytes.

Name	Size
Bit (b)	A single binary digit (1 or 0)
Nibble	4 bits
Byte (B)	8 bits
Kilobyte (kB)	1000 bytes
Megabyte (MB)	1000 kilobytes
Gigabyte (GB)	1000 megabytes
Terabyte (TB)	1000 gigabytes
Petabyte (PB)	1000 terabytes

You might see each unit defined to be 1024 (not 1000) times bigger than the previous unit. The main reason is that 1024 is a power of 2 which is helpful when dealing with binary data.

5) Each bit can take one of two different values (either 1 or 0). This means that a nibble (4 bits) can take 2^4 = 16 different values, and a byte (8 bits) can take 2^8 = 256 different values.

Parity Bits are used to check for Errors

1) Check digits are a way of checking that data has been entered and read correctly. They are digits added to the end of numbers (e.g. ISBNs on books) and are calculated using the other digits in the number. If the check digit is correct when data is read, then it's likely the data has been entered/read correctly.
2) For binary data, the check digit is called a parity bit. You can have even and odd parity bits:
   - An even parity bit is added to make a binary string have an even number of 1s.
   - An odd parity bit is added to make a binary string have an odd numbers of 1s.

The 7-bit string 1010010 has three 1s, so the even parity bit is 1. This is sent as 10100101.
The 7-bit string 1100101 has four 1s, so the even parity bit is 0. This is sent as 11001010.

3) If one bit of the binary string is read incorrectly then the computer will pick up on the error. E.g. if an even parity bit is used and 10101101 is read.
4) If two bits within the same binary string are read incorrectly then no error will be detected. For example, if an even parity bit is used and the binary string 10100000 is read as 10000100 then no error will be picked up.

If odd parity bits were used then these strings would be 10100100 and 11001011.

This page has me in bits...

Keep working your way through that unit table until the size order is clear in your head — it might just show up on your exam. A bit is smaller than a nibble, and a nibble is less than a full byte. I know, hilarious.

Warm-Up and Worked Exam Questions

Try these warm-up questions on logic and units — if there is anything you're unsure about, have a look back through the first part of this section before having a go at the exam questions on the next page.

Warm-Up Questions

1) Look at the three logic expressions in the box below:

(NOT A) AND B	C OR (NOT D)	(E AND F) OR G

a) Draw a logic circuit for each expression.
b) Construct a truth table for each expression.

2) Put these units in order of size: Gigabyte, Kilobyte, Nibble, Megabyte, Byte

3) Look at the three strings below:

String A: 0110101	String B: 0100101	String C: 011010010100101

a) Add an even parity bit to each string.
b) Add an odd parity bit to each string.

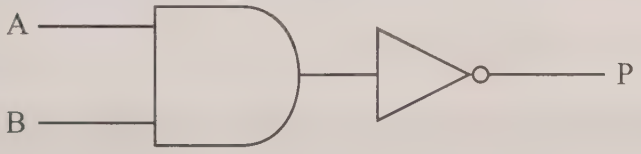
Worked Exam Questions

1 A logic gate can be written as P = A AND B.

a) State the value of input B when input A is 1 and output P is 0.

B = ...0....
[1 mark]

b) A NOT logic gate is placed after the AND logic gate to make the logic diagram below. State the input values when output P is 0.



For the NOT gate to output 0, the input of the NOT gate must be 1.

A = ...1.... B = ...1....
[1 mark]

2 Misha wants to save some music files onto a solid state drive (SSD).

a) State which SSD has the largest capacity:
250 gigabyte (GB), 200 000 megabyte (MB) or 0.3 terabyte (TB).

Convert all the values to the same unit — then compare them.

0.3 Terabyte = 300 GB
200 000 MB = 200 GB

....0.3 Terabyte....
[1 mark]

b) Calculate how many 5 MB music files Misha could save onto a 250 GB SSD.

1 GB = 1000 MB
So 250 GB = 250 × 1000 = 250 000 MB
250 000 ÷ 5 = 50 000 songs

....50 000....
[2 marks]

Exam Questions

3 A series of transistors make the two-level logic circuit (NOT A) AND (B AND C).

a) Complete the truth table below.

A	B	C	NOT A	B AND C	(NOT A) AND (B AND C)
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

[3 marks]

b) Draw the logic diagram that represents (NOT A) AND (B AND C).

Have a look at p.98 for a reminder of what the gates should look like.

[3 marks]

4 Computers process data in binary code and often use check digits.

a) Outline what is meant by a check digit.

.....

.....

[2 marks]

b) Describe how binary is used to represent data in computers.

.....

.....

[2 marks]

c) An even parity bit has been added to the end of three 7-bit binary codes to create the 8-bit binary codes below. Identify and explain which code contains an error.

Code 1	Code 2	Code 3
10101011	10100101	10010011

.....

.....

[2 marks]

d) Explain how a binary code containing a parity bit can be read incorrectly without any errors being detected.

.....

.....

[2 marks]

Binary Numbers

As computers only understand 1s and 0s, all data must be converted into binary to be processed. Binary can be used to represent all numbers in our standard number system.

Counting in Binary is a bit like Counting in Denary

1) In our standard number system we have ten different digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). This is called denary, decimal or base-10.
2) Binary only uses two different digits (0 and 1) — we call this base-2.
3) Counting in binary is similar to counting in denary, but the place values from right to left increase by powers of 2 (e.g. 8, 4, 2, 1), instead of powers of 10 (e.g. 1000, 100, 10, 1).
4) The following table shows the binary equivalents of the denary numbers 0-15:

0 = 0	4 = 100	8 = 1000	12 = 1100
1 = 1	5 = 101	9 = 1001	13 = 1101
2 = 10	6 = 110	10 = 1010	14 = 1110
3 = 11	7 = 111	11 = 1011	15 = 1111

Binary Numbers are easier to Convert using Tables

Drawing a table with binary place values in the first row makes binary to denary conversion easier.

EXAMPLE: **Convert the 8-bit binary number 0011 0101 to a denary number.**

1) Draw up a table with binary place values in the top row. Start with 1 at the right, then move left, doubling each time.
2) Write the binary number 0011 0101 into your table.
3) Add up all the numbers with a 1 in their column:

128	64	32	16	8	4	2	1
0	0	1	1	0	1	0	1

Each column is just a power of 2. i.e. 2^7, 2^6, 2^5, 2^4, 2^3, 2^2, 2^1, 2^0

32 + 16 + 4 + 1 = 53. So 00110101 is 53 in denary.

This works with all binary numbers — just draw as many columns as you need, doubling each time.

8-bit numbers can represent the denary numbers 0 to 255. 16-bit numbers can show the numbers 0 to 65535, and 32-bit can show the numbers 0 to 4294967295.

Convert Denary to Binary by Subtracting

When converting from denary to binary, it's easier to draw a table of binary place values, then subtract them from largest to smallest. Have a look at this example:

EXAMPLE: **Convert the denary number 79 into an 8-bit binary number.**

1) Draw an 8-bit table.
2) Move along the table, only subtracting the number in each column from your running total if it gives a positive answer.
3) Put a 1 in every column that gives a positive answer, and a 0 in the rest.

128	64	32	16	8	4	2	1
0	1	0	0	1	1	1	1

79 − 128 = −49 (no) · 79 − 64 = 15 · 15 − 32 = −17 (no) · 15 − 16 = −1 (no) · 15 − 8 = 7 · 7 − 4 = 3 · 3 − 2 = 1 · 1 − 1 = 0

So 79 converted to an 8-bit binary number is 01001111.

There are other methods to convert denary to binary, so just choose the one you are most comfortable with.

Use powers of 2 to convert between binary and denary...

There's a really easy way to test yourself on this stuff. Write down a denary number between 0 and 255 and convert it to binary. Then write down an 8-bit binary number and convert it to denary.

Binary Numbers

Add Binary Numbers using Column Addition

As binary only uses 1s and 0s we can comfortably do 0 + 0 = 0, 1 + 0 = 1 and 0 + 1 = 1.
Using binary we can't write 1 + 1 = 2. Instead, we have to write 1 + 1 = 10.

EXAMPLES: **1. Add the following 8-bit binary numbers together: 10001101 and 01001000**

1) First, put the binary numbers into columns.
2) Starting from the right, add the numbers in columns.
3) When doing 1 + 1 = 10, carry the 1 into the next column.

```
  1 0 0 0 1 1 0 1
+ 0 1 0 0 1 0 0 0
-----------------
  1 1 0 1 0 1 0 1
    1
```

So 10001101 + 01001000 = 11010101

2. Add the two 8-bit binary numbers below:

```
  0 0 1 1 0 0 1 1
+ 0 1 1 1 1 0 0 1
-----------------
  1 0 1 0 1 1 0 0
  1 1 1     1 1
```

1) Start at the right-hand side and add each column.
2) Sometimes you'll get something like 1 + 1 + 1 = 11, so you need to write 1, then carry 1 to the next column.

So 00110011 + 01111001 = 10101100

You can check your answer by converting the numbers and answer to denary, to make sure it still works.

Overflow Errors occur when a Number has Too Many bits

1) Sometimes, during binary arithmetic you will get a result that requires more bits than the CPU is expecting — this is called overflow.
2) For example, in binary the 8-bit calculation 1111 1111 + 0000 0001 gives the 9-bit answer 1 0000 0000. Computers will see the 1 as an overflow error and just output 0000 0000, which is nonsense.
3) Computers usually deal with these extra bits by storing them elsewhere.
4) Overflow flags are used to show that an overflow error has occurred.

EXAMPLE: **a) Add the 8-bit binary numbers below, giving your answer as an 8-bit binary number.**

1) Add the binary numbers in the usual way.
2) The final calculation is 1 + 1 = 10, so carry the 1.
3) You are left with a 9-bit answer — this is an overflow error.
4) Ignore the overflow to give your 8-bit answer. 01100101

```
  1 1 0 1 0 0 0 1
+ 1 0 0 1 0 1 0 0
-----------------
1 0 1 1 0 0 1 0 1
1       1
```

b) Identify any problems that could be caused by giving your answer as an 8-bit number.

There is an overflow error which can lead to a loss of data and a loss of accuracy in your answer. It could also cause software to crash if it doesn't have a way of dealing with the extra bit.

Binary, binary, quite contrary, how do you overflow...

Overflows occur when a calculation gives a result with more bits than are available to store it. This can be a real problem — programmers must make sure that they can't occur, or that they are dealt with.

Binary Numbers

Binary Shifts can be used to Multiply or Divide by 2

1) A binary shift (also known as a logical shift) moves every bit in a binary number left or right a certain number of places.
2) Gaps at the beginning or end of the number are filled in with 0s.
3) The direction of the binary shift indicates whether it multiplies or divides the binary number:

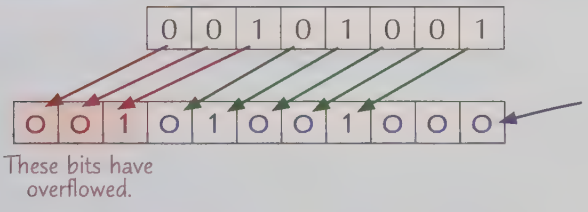
Left shifts MULTIPLY a binary number.
For every place shifted left, the number is doubled.

Right shifts DIVIDE a binary number.
For every place shifted right, the number is halved.

4) If a number is shifted 3 places right, it would be halved three times (i.e. divided by $2^3 = 8$).
If a number were shifted 4 places left, it would be doubled four times (i.e. multiplied by $2^4 = 16$).
5) Left shifts can cause overflows (if extra bits are needed), and right shifts cause bits to 'drop off' the end. Bits dropping off or overflowing can lead to a loss of accuracy/data.

Examples of Binary Shifts

EXAMPLE: **Perform a 3 place left shift on the 8-bit binary number 0010 1001. Explain the effect this will have on the number and problems that may occur.**



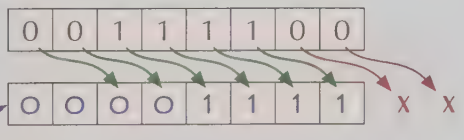
1) Write down the original binary number, then shift all digits 3 places to the left.
2) Fill in the gaps on the right with 0s.

The number has been doubled three times, so it has been multiplied by $2^3 = 8$.

If there are only 8 bits available to store the number then there is an overflow. Some data/accuracy may be lost and an overflow flag will be displayed.

EXAMPLE: **Perform a 2 place right shift on the binary number 0011 1100. What effect will this have on the number?**

1) Write down the original binary number, then shift all digits 2 places to the right.
2) Fill in the gaps on the left with 0s.



A 2 place right shift gives the binary number 0000 1111. As this is a 2 place shift, the original number will have been halved twice (so divided by $2^2 = 4$).

Dividing using a right binary shift has the same effect as using the DIV operator (p.64) — your answer would not take into account any remainders.

Shift left to multiply and shift right to divide...

Binary shifts are really good for doing fast multiplication and division by powers of 2. Watch out for bits overflowing or dropping off the end — if this happens then you'll lose accuracy in your answer.

Hexadecimal Numbers

Hexadecimal (hex) is another number system used regularly in programming. Hex uses a combination of digits and letters in order to represent a number.

Hexadecimal numbers are Shorter than Binary

1) Hexadecimal (or base-16) uses sixteen different digits.
2) A single hex character can represent any denary number from 0-15. To represent 0-15 in binary would require 4 bits (a nibble), so each hex character equates to a nibble in binary.
3) The table shows the denary and binary value of each hex character.
4) Programmers often prefer hex when coding, because:

- It's simpler to remember large numbers in hex — they're far shorter than binary numbers.
- Due to hex numbers being shorter, there's less chance of input errors.
- It's easier to convert between binary and hex than binary and denary.

Computers themselves do not use hex — they still have to convert everything to binary to process it.

Denary	Hex	Binary	Denary	Hex	Binary
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100
5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111

Convert Hex to Denary by Multiplying each Character

In hex, moving right to left, place values increase in powers of 16.

4096	256	16	1

To convert from hex to denary, draw up a table, fill in the boxes, then multiply — just like in this example:

EXAMPLES:

1. Convert the hexadecimal number 87 into denary.

1) First, draw this table, then write in your hex number.

16	1
8	7

2) Multiply the numbers in each column. 8 × 16 = 128 7 × 1 = 7
3) Add up the results: 128 + 7 = 135

So the hex number 87 is 135 in denary.

Luckily in the exam you'll only have to convert two digit hex numbers like in these examples.

To convert from denary to hex, draw the table but use division to fill it in.

2. Convert the denary number 106 into hexadecimal.

Remember, hex goes from 0-9, then A to F.

1) Start at the left. Divide 106 by 16, then hold onto the remainder. 106 ÷ 16 = 6 r 10

16	1
6	A

2) Divide the remainder from the last calculation by 1. 10 ÷ 1 = 10 = A

So the denary number 106 is 6A in hexadecimal.

REVISION TASK

Hex can be a blessing and a curse...

Hex and denary can look fairly similar (as they both contain 0-9), so make sure you've got them the right way round when converting — 65 in hexadecimal is NOT the same as 65 in denary. Memorise the hex table and its advantages, then cover it up and write everything you remember.

Hexadecimal Numbers

Convert **Binary** to **Hex** by splitting it into **Nibbles**

1) Each hex character is equal to a nibble in binary, so it is possible to convert from binary to hex by splitting the binary code into 4-bit chunks.
2) Binary to hex conversions can be much easier than converting from binary to denary, as you only have to deal with the nibbles one at a time.

EXAMPLE: **Convert the binary number 1011 1001 to hexadecimal.**

Remember, hex only uses letters for denary values between 10-15.

1) Firstly, split the binary number into nibbles: 1011 1001
2) Draw a table with columns labelled 1, 2, 4, 8, then repeat the values for as many nibbles as you require.
3) Fill in the table with your binary number.
4) For each nibble, add up the numbers with a 1 in the column, then convert this value to hex.
5) Finally put the hex values together.

8	4	2	1	8	4	2	1
1	0	1	1	1	0	0	1

8 + 2 + 1 = 11 = B 8 + 1 = 9

The binary number 10111001 is B9 in hexadecimal.

If the binary number can't be split into nibbles, you'll have to stick some zeros on the front.

EXAMPLE: **Convert the binary number 1 1111 01000 to hexadecimal.**

1) Add zeros to the front of the binary number, so that you can split it into nibbles. 0011 1110 1000
2) Draw a repeating table of 1, 2, 4 and 8, as above.
3) Write your binary number in the table.
4) Add up each nibble and convert each value to hex.
5) Put the hex values together.

8	4	2	1	8	4	2	1	8	4	2	1
0	0	1	1	1	1	1	0	1	0	0	0

2 + 1 = 3 8 + 4 + 2 = 14 = E 8 = 8

The binary number 111110 1000 is 3E8 in hexadecimal.

For **Hex** to **Binary**, use each **Character's Denary Value**

To convert the opposite way (from hex to binary) convert each hex character into binary, then just put the binary numbers together.

EXAMPLE: **Convert the hexadecimal number 8C to binary.**

1) First, find the denary value of each character: 8 = 8 in denary, C = 12 in denary
2) Find the binary value of each denary number:
3) Put the nibbles together to get the equivalent binary number.

8	4	2	1
1	0	0	0

8 = 1000 in binary

8	4	2	1
1	1	0	0

12 = 1100 in binary

The hexadecimal number 8C is 10001100 in binary.

REVISION TIP

This page has so many nibbles it could spoil your lunch...

When converting binary to hex just remember to split the binary up into chunks of 4 bits, each with columns labelled 1, 2, 4 and 8. Then work out the individual hex values and put them together.

Warm-Up and Worked Exam Questions

Converting between binary, denary and hexadecimal tests your maths skills just as much as your computing knowledge. Have a go at these warm-up questions before diving into the exam questions on the next page.

Warm-Up Questions

1) Convert:
   a) 01011001 from binary to denary
   b) 69 from denary to binary
2) Add the following 8-bit binary numbers together: 01101000 and 10001010
3) Explain how overflow errors occur.
4) Perform a 3 place right shift on 10111000. Give your answer as an 8-bit number.
5) What is the effect of a 1 place left shift?
6) Convert:
   a) 48 from denary to hex
   b) C6 from hex to denary
   c) 10100010 from binary to hex
   d) 3D from hex to binary

Worked Exam Questions

1 Work out these conversions.

a) Convert the 8-bit binary number 10010011 into a denary number.

128 + 16 + 2 + 1 = 147

147

[1 mark]

b) Convert the denary number 252 into an 8-bit binary number.

Drawing an 8-bit table can help.

128	64	32	16	8	4	2	1
1	1	1	1	1	1	0	0

252 − 128 = 124
124 − 64 = 60
60 − 32 = 28
28 − 16 = 12
12 − 8 = 4
4 − 4 = 0

11111100

[1 mark]

2 Daniel is a programmer. He makes the following two claims about hex numbers.

Claim 1: "Hex is much easier to work with than binary."

Claim 2: "Converting from denary to hex is easier than converting from binary to hex."

Would other programmers agree with Daniel's claims? Explain your answers.

Claim 1: They're likely to agree with claim 1. Hex numbers are shorter so are easier to identify, remember, edit and share than binary codes.

Claim 2: They're likely to disagree with claim 2. Converting binary to hex is easier as binary numbers can be split into nibbles to quickly read off the hex values.

[4 marks]

Exam Questions

3 Binary shifts can be used to quickly multiply and divide binary numbers.

a) Complete a 3 place left shift on the binary number 00011010.

.....

[1 mark]

b) State an appropriate binary shift to divide a binary number by 4 and use it on 11010100.

.....

[2 marks]

c) Yasha says "Adding a binary number to itself is the same as a 2 place left shift."
Is he correct? Explain your answer.

.....

.....

[2 marks]

4 A security program encrypts passwords using a hexadecimal conversion.
The binary code of each letter for the password 'CAT' is shown below.

01000011 01000001 01010100

a) Convert each binary number above to a hexadecimal number to encrypt the password 'CAT'.

.....

[3 marks]

b) The password 'DOG' is encrypted as 44 4F 47.

i) Convert the first encrypted letter to binary.

.....

[1 mark]

ii) What password would be encrypted as 43 4F 44 45?

Look back at previous question parts.

.....

[2 marks]

5 The function `denary()` converts one hexadecimal character to denary. (e.g. `denary(F)` = 15).

a) Calculate the value of `denary(A)` + `denary(C)`.

.....

[2 marks]

b) Write an algorithm using `denary()` to convert any 2-digit hexadecimal into denary.

Think about splitting the 2-digit hexadecimal into separate characters.

[4 marks]

Characters

Almost everything can be represented as binary code — words, images and sound can all be turned into bits and processed by a computer. Firstly let's look at words, which are made up of different characters.

Binary can be used to represent Characters

1) Alphanumeric characters are used to make words and strings (see p.67). They include uppercase and lowercase letters, the digits 0-9, and symbols like ? + and £.
2) Computers are unable to process these characters directly as they only process binary code. So they need a way of converting these characters to binary code and vice versa. They can do this using character sets.

> Character sets are collections of characters that a computer recognises from their binary representation.

Don't mistake a character set for a font. A character set is what determines the letter — the font you use just displays that letter in a certain way.

3) As well as the alphanumeric characters mentioned above, character sets also contain special characters which do certain commands (e.g. space, enter and delete).
4) So when you press a button on your keyboard it sends a binary signal to the computer telling it which key you pressed. The computer then uses the character set to translate the binary code into a particular character.

The number of Bits you'll need is based on the Character Set

Different character sets can have different amounts of characters. The number of characters in a character set determines how many bits you'll need. Here are some standard character sets you should know about:

ASCII

- ASCII is the most commonly-used character set in the English-speaking world. Each ASCII character is given a 7-bit binary code — this means it can represent a total of 128 different characters, including all the letters in the English alphabet, numbers, symbols and commands.
- An extra bit (0) is added to the start of the binary code for each ASCII character (see the table on the right). This means each ASCII character fits nicely into 1 byte.

Character	Binary	Hex	Denary
Backspace	00001000	8	8
+	00101011	2B	43
3	00110011	33	51
B	01000010	42	66
b	01100010	62	98

Some examples of ASCII characters.

Extended ASCII

- Extended ASCII is a character set which gives each character an 8-bit binary code, allowing for 256 characters to be represented. The first 128 characters are in exactly the same order as the ASCII characters.
- Extended ASCII is particularly useful for many European languages like French and German which include accents on some of the vowels, like é, ô and ü.

Unicode®

- Unicode® comes in several different forms and tries to cover every possible character that might be written. In its most common forms it uses 16-bit and 32-bit binary codes.
- The best thing about Unicode® is that it covers all major languages, even those that use a completely different alphabet like Greek, Russian and Chinese.

Character sets are used to turn binary data into characters...

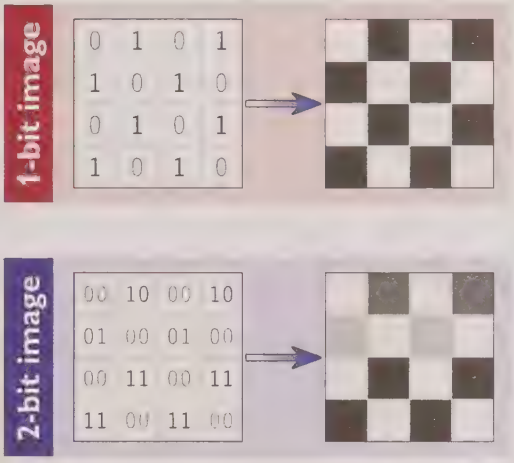
Extended ASCII comes in several different forms. The first 128 characters are always the same, but the ones after that can change to suit the language you're writing in. E.g. the "Latin 1 Western European" character set is used for languages like French and German, while another is better suited to Eastern European languages.

Storing Images

Images and sounds are pieces of data stored on computers — so, naturally, they're made of bits (p.100).

Images are stored as a series of Pixels

1) The type of images you use most often are called bitmap images — they're mainly used for photos. Bitmap images are made up of lots of tiny dots, called pixels.
2) The colour of each pixel is represented by a binary code. The number of colours available in an image is related to the number of bits the code has.
3) Black-and-white images only use two colours, meaning they only need 1-bit to represent each pixel — 0 for white and 1 for black.
4) 2-bit images can be made up of four colours. Each pixel can be one of four binary values — 00, 01, 10 and 11.
5) You can make a greater range of shades and colours by increasing the number of bits for each pixel.



Increasing Colour Depth and Resolution increases the File Size

1) The colour depth is the number of bits used for each pixel.
2) Given the colour depth you can work out how many colours can be made using this formula:

Total number of colours = 2^n (where n = number of bits per pixel, or bpp)

1-bit image: $2^1 = 2$ colours | 4-bit image: $2^4 = 16$ colours | 24-bit image: $2^{24} = 16\,777\,216$ colours

3) Most devices use a 24-bit colour depth, with 8 bits used to indicate the levels of red, green and blue needed for each pixel. It's estimated that the human eye can see around 10 million different colours, so a 24-bit colour depth should cover every colour that you could possibly see.
4) The resolution is the density of pixels in an image, i.e. how many pixels are within a certain area. It's normally measured in dots per inch (dpi).
5) The higher the resolution, the more pixels in a certain area and so the better quality of image. E.g. if an image has a resolution of 60 dpi, it means that a one-inch square contains a grid that is 60 pixels wide and 60 pixels high. So there are 60 x 60 = 3600 pixels in that square inch. If we increased the resolution to 90 dpi, it would mean 8100 pixels in that square inch, so a better image quality.
6) Increasing the resolution or the colour depth means that there are more bits in the image. This improves the image quality, but also increases the file size.

Devices need Metadata to display the images

1) Metadata is the information stored in an image file which helps the computer recreate the image on screen from the binary data in each pixel.
2) Metadata usually includes the image's file format, height, width, colour depth and resolution. It can also include extra information, like the time and date that the image was created or last edited.
3) Without metadata, devices would not be able to display the image on screen as intended.

Images are stored as long strings of bits...

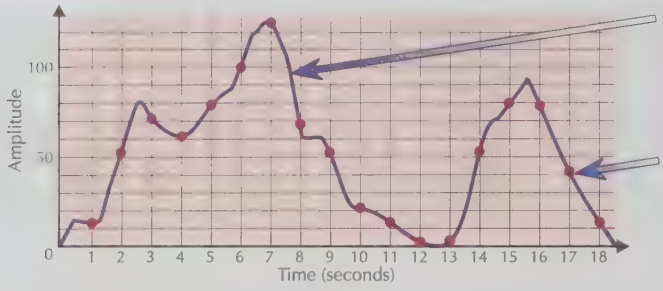
You might see the term resolution used to mean the dimensions of an image in pixels (e.g. 720 × 1280), but don't get misled — if you're asked about it in the exam, they'll be talking about pixel density in dpi.

Storing Sound

Sound is made up of bits and stored in files on a computer. Or rather, digital sound is — the other type of sound, analogue, doesn't get on well with computers very much, so we've got to turn it into digital first.

Sound is Sampled and stored Digitally

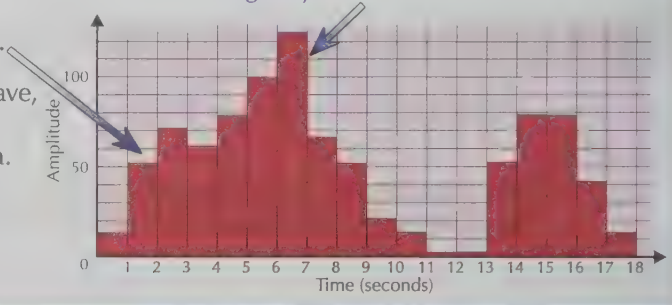
1) Sound is recorded by a microphone as an analogue signal. Analogue signals are pieces of continually changing data.
2) Analogue signals need to be converted into digital data so that computers can read and store sound files. This is done by analogue to digital converters, which are found in most modern recording devices.
3) The process of converting analogue to digital is called sampling:



The blue line shows the analogue sound wave — it's one continuous piece of data which keeps changing.

To convert the analogue recording to digital data, we sample the amplitude of the wave at regular intervals (shown by the dots on the graph). The amplitude can only take certain values depending on the bit rate (see below).

Once the device has sampled the recording, it creates the curve digitally like this.



Each block of data matches where each sample was taken.

The digital data is about the same shape as the analogue wave, but it's not continuous. It's lost a lot of data — e.g. the last peak in the analogue wave is much flatter in the digital data.

The digital data can be improved by taking samples more regularly — most music isn't sampled every second but every couple of milliseconds.

Several factors affect the Size and Quality of Sound Files

1) Sampling intervals are the gaps between each of the points where the analogue recording is sampled. E.g. the audio file might be sampled every 5 milliseconds (ms) — the sampling interval would be 5 ms.
2) Sampling frequency (or sample rate) is how many samples you take in a second — it's usually measured in kilohertz (kHz). E.g. a common sampling frequency is 44,100 samples per second (44.1 kHz).
3) Sample size is the number of bits available for each sample (like colour depth but for sound samples).
4) Bit rate is the number of bits used per second of audio — it's calculated using this formula:

Bit rate = Sampling frequency × sample size

The bit rate is usually measured in kilobits per second (kbit/s).

5) Increasing the sampling frequency means the analogue recording is sampled more often. The sampled sound will be better quality and will more closely match the original recording.
6) Increasing the sample size means the digital file picks up quieter sounds, even if they're happening at the same time as louder ones. This will also result in a sampled sound that is closer to the quality of the original recording.
7) However, increasing the sampling frequency and sample size will increase the bit rate. This will increase the number of bits in the sound file, which means a larger file size.

You can change the size and quality of sound files...

REVISION TIP

The factors which impact the size and quality of sound files are all related — e.g. increasing the sampling frequency means more samples are taken per second, which decreases the sampling interval. Similarly, the sampling frequency and sample size are multiplied together to calculate the bit rate, so increasing either of them will increase the bit rate, and the file size too.

Compression

In the modern world, we're practically swimming in badly lit photos and subpar pop songs — so many, in fact, that you'd start to wonder how we can possibly store them all. The answer is down to data compression.

Sometimes we need to **Compress** files

1) Data compression is when we make file sizes smaller, while trying to make the compressed file as true to the original as possible.
2) Compressing data files has many uses:

- Smaller files take up less storage space on a device.
- Streaming and downloading files from the Internet is quicker as they take up less bandwidth (p.20).
- It allows web pages to load more quickly in web browsers.
- Email services normally have restrictions on the size of the attachment you can send — compressing the file allows you to send the same content with a much smaller file size.

There are **Two Types** of compression — **Lossy** and **Lossless**

1) Lossy compression works by permanently removing data from the file — this limits the number of bits the file needs and so reduces its size.
2) Lossless compression makes the file smaller by temporarily removing data to store the file and then restores it to its original state when it's opened.

	Pros	Cons	E.g. of File Types
Lossy	• Greatly reduced file size, meaning more files can be stored. • Lossy files take up less bandwidth so can be downloaded and streamed more quickly. • Commonly used — lots of software can read lossy files.	• Lossy compression loses data — the file can't be turned back into the original. • Lossy compression can't be used on text or software files as these files need to retain all the information of the original. • Lossy files are worse quality than the original. But, this loss in quality is normally unnoticeable.	• MP3 (audio) • AAC (audio) • JPEG (image)
Lossless	• Data is only removed temporarily so there is no reduction in quality — the compressed file should look or sound like the original. • Lossless files can be decompressed — turned back into the original. • Lossless compression can be used on text and software files.	• Only a slight reduction in file size, so lossless files still take up quite a bit of space on your device. E.g. a lossless song may have a file size of around 30 MB, while the same song with lossy compression may be 5 MB.	• FLAC (audio) • TIFF (image) • PNG (image)

EXAMPLE:

Phil has just heard a new band on the radio. He wants to download fifty of their songs from the Internet and store them on his smartphone to take on holiday. State which type of compression would be most appropriate in this situation and explain why.

Lossy compression would be the most appropriate. Lossy files are smaller so they would take up less bandwidth, meaning Phil could download the songs more quickly. Their smaller file size would also allow him to store them all on his smartphone without taking up too much storage space.

The best compression type? I'm afraid I'm at a loss...

Lossy files aren't as high quality as the originals, but the difference is normally unnoticeable to us unperceptive humans. This helps to explain why lossy file formats like JPEG (for photos) and MP3 (for music) are so popular — they save a lot of storage space and their inferior quality is hardly noticeable.

Warm-Up and Worked Exam Questions

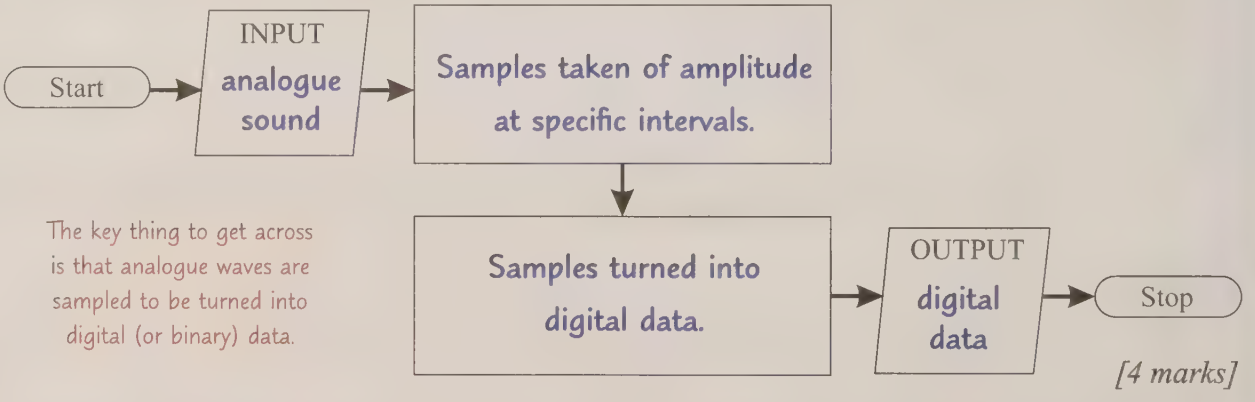
That's the last of the stuff you have to learn, but you're not finished yet — there are a whole bunch of questions to answer. For the final time, test what you've learnt with these warm-up and exam questions.

Warm-Up Questions

1) What is the difference between the character sets ASCII and Unicode®?
2) How many different colours are possible with a colour depth of 4?
3) What would be the effect on the file size of an image if:
   a) the colour depth was increased? b) the resolution was decreased?
4) What is the bit rate of an audio file with a sampling frequency of 50 kHz and a sample size of 4 bits?
5) What would be the effect on the file size and quality of an audio sample if:
   a) the bit rate was increased? b) the sampling interval was increased?
6) Name the two types of compression and give one reason why each one is useful.

Worked Exam Questions

1 Sound is converted from an analogue sound wave to a digital format using sampling. Complete the flow diagram below to show the process of sampling.



The key thing to get across is that analogue waves are sampled to be turned into digital (or binary) data.

[4 marks]

2 State and explain which type of compression would be most appropriate in these examples.

a) Uploading 100 holiday photographs to a social media account.

Make sure the explanation relates to the context of the question.

Type Of Compression: Lossy compression

Explanation: Lossy compression will greatly reduce the size of the files so uploading them will be quicker.

[2 marks]

b) Uploading a photograph of a model for a fashion magazine.

Type Of Compression: Lossless compression

Explanation: An image with all original detail may be preferred so they can edit it and print it at the best possible resolution.

[2 marks]

Exam Questions

3 Helena is writing a news article using a word processor.

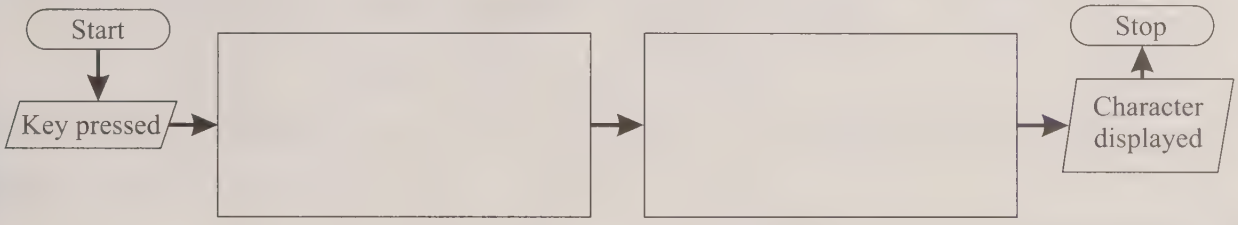
a) Define the term 'character set'.

.....

.....

[1 mark]

b) Complete the flowchart to explain how Helena's computer recognises characters she enters.



[2 marks]

4 Jade records herself reading two extracts from a novel to use in an audiobook. The bit rates and sampling frequencies of each recording are shown below.

	Length	Bit Rate	Sampling Frequency
Extract 1	2 minutes	128 kbit/s	32 kHz
Extract 2	2 minutes	320 kbit/s	44.1 kHz

a) Explain which extract would have the better sound quality.

.....

.....

[2 marks]

b) Give **one** drawback of using extract 2 rather than extract 1 for the audiobook.

.....

[1 mark]

5 Duncan prints a 10 × 10 inch photograph with a resolution of 60 DPI.

a) Calculate the total number of pixels in Duncan's photograph.

Remember that you're not allowed to use a calculator.

.....

[2 marks]

b) Explain how decreasing the DPI would affect the image quality.

.....

.....

[2 marks]

c) Explain the purpose of metadata in an image file.

.....

.....

[2 marks]

Revision Questions for Section Seven

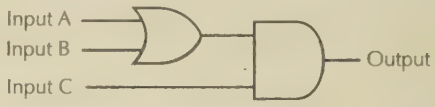
That's the end of the content. Get through these questions, then it's time to have a go at the practice papers.

- Try these questions and tick off each one when you get it right.
- When you've done all the questions for a topic and are completely happy with it, tick off the topic.

Logic and Units (p.98-100) ☐

1) For each of the 3 main logic gates: a) Draw its symbol.
   b) State how many inputs and outputs it has.
   c) Draw its truth table.

2)* Draw the truth table for the logic diagram on the right.



3) Why is binary used by computers?

4) Put these units in order of size: Terabyte, Petabyte, Kilobyte, Gigabyte, Megabyte

5)* A hard drive has a storage capacity of 2000 gigabytes.
   a) How many terabytes is this? b) How many megabytes is this?

Binary and Hexadecimal (p.103-107) ☐

6)* Convert the following denary numbers to: a) binary b) hexadecimal
   (i) 17 (ii) 148 (iii) 240

7)* Convert the following binary numbers to: a) denary b) hexadecimal
   (i) 00111000 (ii) 10011111 (iii) 101011

8)* Convert these hexadecimal numbers to: a) denary b) binary
   (i) 4A (ii) 75 (iii) BD9

9)* Add the binary numbers 01011101 and 00110010.

10) What is an overflow error?

11) What effect do left and right shifts have on binary numbers?

12) Give three reasons why programmers prefer hexadecimal over binary and denary.

Characters (p.110) ☐

13) What is the definition of a character set?

14) Give the four types of character that are included in a character set.

15) a) What are the three main character sets?
   b) For each of your answers to part a) state how many bits it takes to represent each character.

Images, Sound & Compression (p.111-113) ☐

16) What is meant by a bitmap image?

17) Define colour depth.

18) What is an image's resolution and what units do we use to measure it?

19) Name two ways in which increasing the resolution or colour depth will affect the image.

20) What is metadata and what is it used for?

21) In no more than four bullet points, explain how audio sampling works.

22) Give a definition for each of the following and explain what happens when you increase each of them:
   a) sampling frequency b) sampling interval c) sample size d) bit rate

23) Give four reasons why you might want to compress data.

24) What is the difference between lossy compression and lossless compression?

25) Give three reasons why you might want to use: a) lossy compression b) lossless compression

*Answers on p.147

Practice Paper 1

Once you've been through all the questions in this book, you should be starting to feel prepared for the final exams. This practice paper will test you on Sections 1-3 of this book and contains a mix of short answer and longer answer questions. It also has an extended writing question and an algorithm question.

GCSE OCR Computer Science

Practice Paper 1
Computer Systems

Centre name					
Centre number					
Candidate number					

Time allowed:
- 1 hour 30 minutes

Surname Doublet
Other names Victor
Candidate signature

You **may not** use a calculator

Instructions to candidates
- Write your name and other details in the spaces provided above.
- Answer **all** questions in the spaces provided.
- Do all rough work in this book. Cross through any work you do not want to be marked.

Information for candidates
- There are 80 marks available on this paper.
- The marks available are given in brackets at the end of each question.
- Quality of extended responses will be assessed in this paper in questions marked with an asterisk (*).

For examiner's use							
Q	Attempt N°			Q	Attempt N°		
1				5			
2				6			
3				7			
4							
			Total				

Answer **all** questions in the spaces provided

1. Annie has a three year old laptop. She is giving it a full service before selling it on.

(a) Annie runs some 'Disk Health' utility software to check for any problems with her HDD.

(i) Define what is meant by utility software.

Software designed to maintain a computer system

[1 mark]

(ii) Give **two** other examples of utility software.

1 Anti-virus software

2 File orginisation software

[2 marks]

(b) The utility reports that Annie's hard disk is 25% fragmented.

(i) Explain **one** problem caused by a fragmented hard disk.

When a hard disk is fragmented it can take longer to read/write data and in turn may slow down the computer

[2 marks]

(ii) Briefly describe the defragmentation process.

Defragmentation software reduces fragmentation by moving files on the hard disk, empty spaces are collected together. This means the read/write heads don't have to move as far across a disk increasing the read/write speed. *[3 marks]*

(c) Annie also plans to perform a disk clean-up. Suggest why it could be better to do the disk clean-up before defragmentation, rather than afterwards.

Disk clean up removes files which would lead to fragmentation.

[1 mark]

(d) Annie is considering getting a new laptop with an SSD rather than an HDD. Give **three** advantages of choosing an SSD over an HDD.

1 An SSD has no moving parts so is more durable

2 SSDs have quicker read/write speeds that HDDs

3 ~~More storage for small form factor in an SSD~~

SSDs make no noise

[3 marks]

2. Karen stores her holiday pictures in the cloud. She decides to download an image from the cloud to her laptop.

(a) Define what is meant by the cloud.

The cloud is server storage provided by a hosting company

[1 mark]

(b) Karen's laptop and the cloud server have a client-server relationship. Describe the communication that takes place between the cloud server and Karen's laptop when she downloads the image.

Karen's laptop connects to the server over internet and fetches the file that is then downloaded onto her laptop

[2 marks]

(c) The image is transferred from the cloud server to Karen's laptop using packets.

(i) Explain how packets are used to transfer the image over the Internet to Karen's laptop.

Karen's laptop sends a request to the server. The server processes the request and responds with the image.

[6 marks]

(ii) Explain why packet switching is an efficient way to send data over large networks.

[2 marks]

(iii) State the name of the network protocol that is responsible for directing packets during the packet switching process.

[1 mark]

(d) The cloud hosting company uses a system of network forensics as part of its network policy.

(i) Define the term network policy.

.....

[1 mark]

(ii) Explain what is meant by network forensics and how they are used.

.....

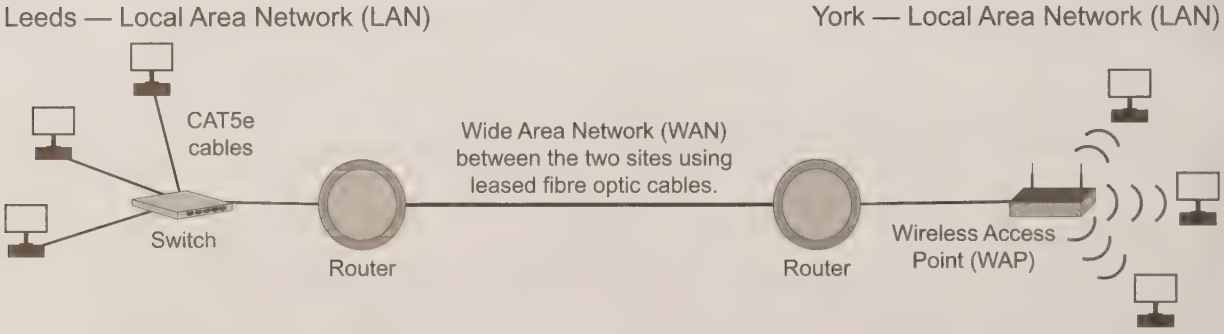
.....

.....

.....

[3 marks]

3. A Yorkshire-based television company has two studios, one based in Leeds and the other based in York. The company's computer network is shown in the diagram below.



(a) The Leeds studio uses wired connections, whereas the York studio uses wireless connections.

(i) Select words from the following list to complete the sentences below:

Ethernet **WPA2** **WAP** **Coaxial** **SQL** **Frame**

..... is a network protocol used on wired networks.

..... is a security protocol used on wireless LANs.

[2 marks]

(ii) Describe **one** difference between a CAT5e twisted pair cable and a coaxial cable.

.....

.....

[2 marks]

(iii) Outline the advantages and disadvantages of each LAN setup.

The Leeds studio's wired setup: .....

.....

.....

The York studio's wireless setup: .....

.....

.....

[4 marks]

(b) The studios are connected in a Wide Area Network (WAN) using fibre optic cables.

(i) State **one** advantage of using fibre optic cables rather than copper cables in a WAN.

.....

[1 mark]

(ii) Identify **one** reason why the company uses leased lines for its WAN.

.....

[1 mark]

4.* Recent years have seen the increasing use of computer technology to distribute and view digital media content (music, movies and books).

Discuss the impact of the increasing use of digital media.

In your answer, you might consider:

- stakeholders
- technology
- environmental issues
- legal issues

.....

.....

.....

.....

[8 marks]

5. Hardeep wants to try a new operating system on his computer.
The new operating system is optimised for use with a touchscreen.

	Hardeep's PC	OS Minimum Requirements
Processor:	2.1 GHz, 4 cores	1.0 GHz, 4 cores
RAM:	2 GB	2 GB
Storage:	256 GB, 125 MB free	19 GB free space
GPU:	Integrated 256 MB	Dedicated 512 MB

(a) Hardeep needs to upgrade some of the components in his computer before the new operating system can be installed. State which components must be upgraded.

1 .....

2 .....

[2 marks]

(b) Would you recommend that Hardeep upgrades any other components in his computer? Explain your answer.

.....

.....

[2 marks]

(c) Explain why an operating system requires a certain amount of RAM.

.....

.....

[2 marks]

(d) The new operating system's GUI is optimised for touchscreen use. Describe **two** features that a GUI may include to take advantage of touchscreen technology.

1 .....

.....

2 .....

.....

[4 marks]

6. Dishley Academy stores personal information about pupils, such as name, age, address and phone number, on their network.

(a) For the following actions, tick the corresponding box to show whether or not they would be allowed under the Data Protection Act 1998.

	Allowed	**Not Allowed**
Giving each teacher a USB flash drive containing the personal information of all the pupils who have attended the school.		
Transferring pupils' personal information to their new school when they leave.		
Refusing to tell a parent what information is being stored about their son/daughter.		
Backing up pupils' personal information on a drive that is stored in a locked safe.		
Using pupils' personal information to get in contact with their parents.		
Putting pupils' personal information on the school website to make it easier for teachers and parents to access.		

[6 marks]

(b) The IT technicians at the academy want to protect pupils' data against possible network attacks. Describe **two** different types of network attack, and suggest a method that could be used to protect against it.

Type of attack: .....

Description: .....

.....

Method of protection: .....

Type of attack: .....

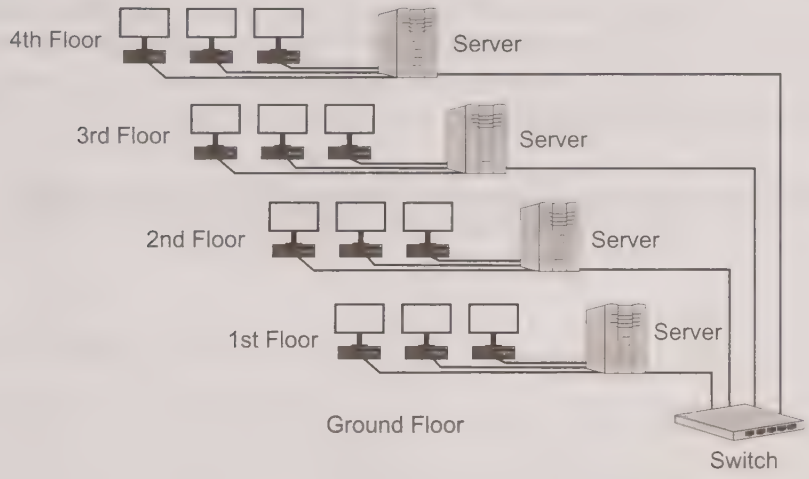
Description: .....

.....

Method of protection: .....

[6 marks]

7. A company has its employees' computers spread across four floors.
The computers on each floor are connected to that floor's server in a star network.
Employees need to access files on all of the servers, so each of the four servers are connected in another star network, with a central switch located on the ground floor.



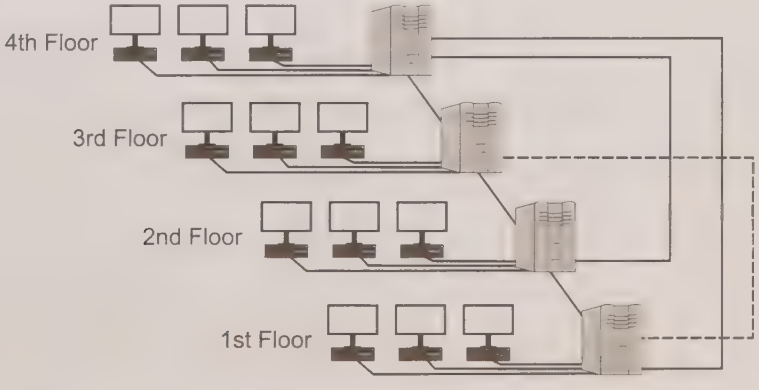
(a) Describe the effect that the central switch failing would have on the rest of the network.

.....

.....

[2 marks]

(b) The company decides to remove the switch from the network and instead connect the four servers in a full mesh network, as shown in this diagram.



Explain the advantages and disadvantages to the company of connecting the servers together in a full mesh network instead of a star network.

.....

.....

.....

.....

.....

[4 marks]

(c) At the end of every day, the servers run a backup utility to decide whether to do an incremental backup, full backup or no backup. The utility uses two different variables:

- `incTotal` stores the number of edited files since the last backup of any kind.
- `fullTotal` stores the number of edited files since the last full backup.

The number of files that have been edited each day are added to both variables. If the value of `incTotal` reaches 200, then the utility takes an incremental backup, and `incTotal` is set to 0. However, if the value of `fullTotal` reaches 1000, then a full backup is taken **instead**, and `fullTotal` and `incTotal` are set to 0.

Write an algorithm to demonstrate the function of the backup utility.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[6 marks]

END OF QUESTIONS

Practice Paper 2

This practice paper will test you on Sections 4-7 of this book and features more algorithm-based questions than Practice Paper 1 — this will be the case in the real exams as well.

GCSE OCR Computer Science

Practice Paper 2
Computational Thinking, Algorithms and Programming

Centre name					
Centre number					
Candidate number					

Time allowed:
- 1 hour 30 minutes

Surname
Other names
Candidate signature

You **may not** use a calculator

Instructions to candidates
- Write your name and other details in the spaces provided above.
- Answer **all** questions in the spaces provided.
- Do all rough work in this book. Cross through any work you do not want to be marked.

Information for candidates
- There are 80 marks available on this paper.
- The marks available are given in brackets at the end of each question.

For examiner's use							
Q	Attempt N°			Q	Attempt N°		
1				6			
2				7			
3				8			
4				9			
5							
			Total				

Answer **all** questions in the spaces provided

1. A petrol station needs a program to calculate the cost of fuel for each customer.

   (a) Identify appropriate data types for:

   (i) The name of the fuel that the customer used.

   .....

   [1 mark]

   (ii) The total cost of the customer's fuel.

   .....

   [1 mark]

   (b) Explain why it's important to use the correct data type to store information.

   .....

   .....

   .....

   [2 marks]

   (c) Describe, with an example, how the function `str()` might be used when printing the receipt.

   .....

   .....

   .....

   [2 marks]

   (d) The source code for the program needs to be translated into machine code. Outline **two** differences in the way a compiler and interpreter would translate the program.

   1 .....

   .....

   2 .....

   .....

   [4 marks]

2. **(a)** Convert the binary number 10101000 into denary.

.....

.....

[2 marks]

(b) Convert the binary number 10110101 into hexadecimal.

.....

.....

[2 marks]

(c) Calculate the sum of the binary numbers 10101000 and 10110101.

.....

.....

[1 mark]

(d) When a computer tried to add the binary numbers in part (c) together, an overflow error occurred. What is meant by the term 'overflow error'?

.....

.....

[1 mark]

3. Lenny is writing a program for an exercise bike that adjusts the difficulty based on information about the user.

(a) One of Lenny's sub programs takes `weight` as a parameter.

(i) Define what is meant by a 'parameter'.

.....

[1 mark]

(ii) What is the scope of a parameter?

.....

[1 mark]

(iii) Outline how arguments are different from parameters.

.....

[1 mark]

(b) The code below appears near the start of Lenny's program.

```
global difficulty
input weight
procedure setInitialDifficulty(weight)
    difficulty = weight DIV 6
endprocedure
```

(i) Explain what the first line of this code does.

.....

.....

[2 marks]

(ii) Describe what the `setInitialDifficulty` procedure does.

.....

.....

[2 marks]

(c) Lenny wants a sub program that adjusts the difficulty depending on the user's heart rate.

- If their heart rate is below 90 then the difficulty level should be increased by 1.
- If their heart rate is over 140 then the difficulty level should be decreased by 1.
- If their heart rate is over 160 then the difficulty level should be set to 0 and the warning message "Slow Down!" should appear.

Write a sub program that takes heart rate as a parameter and adjusts the difficulty level as stated above.

.....

.....

.....

.....

.....

.....

.....

.....

.....

[4 marks]

4. A car uses a logic circuit to decide whether to start the engine or not.

- The car has two buttons, labelled **S** (START) and **D** (DRIVE). If both buttons are on, the engine will start.
- The engine also starts if the ignition switch **I** is turned on.

(a) Draw the logic circuit diagram for this system, with **Z** as an output.

[3 marks]

(b) Fill in the missing values in the truth table for the logic circuit above.

S	D	I	Z
0	1		0
		0	1
1			0

[3 marks]

5. Every 5 minutes, an app on a mobile phone records the electrical current (in mA) passing through a pair of connected headphones. A sample of the readings is shown in the table below.

10 mA	15 mA	12 mA	18 mA	20 mA

(a) Show the stages of a linear search to find the value '12 mA' in the list above.

.....

.....

.....

.....

.....

.....

[2 marks]

(b) If a reading is greater than 30 mA, the mobile phone will buzz and the program will stop. If not, the program will wait 5 minutes before taking another reading. Draw a flow diagram to show this program.

[6 marks]

6. Florence is a graphic designer for a publishing company. The image editing software that she uses represents each unique colour as a six digit hex code.

(a) As a power of 16, how many possible unique colours could Florence use?

.....

[1 mark]

(b) Explain **one** benefit to programmers of using hex codes to represent the different colours.

.....

.....

.....

[2 marks]

Florence saves the same image as three different file types (shown in the table below).

File Type	JPEG	PNG	TIFF
Size	0.2 MB	1 MB	0.9 MB

(c) One of the file types uses lossy compression. State and explain which file type is most likely to be an example of lossy compression.

.....

.....

.....

[2 marks]

(d) Evaluate the impact if Florence always used lossless compression to store all her images.

.....

.....

.....

.....

.....

[4 marks]

(e) Give **two** pieces of metadata that are often included in image files.

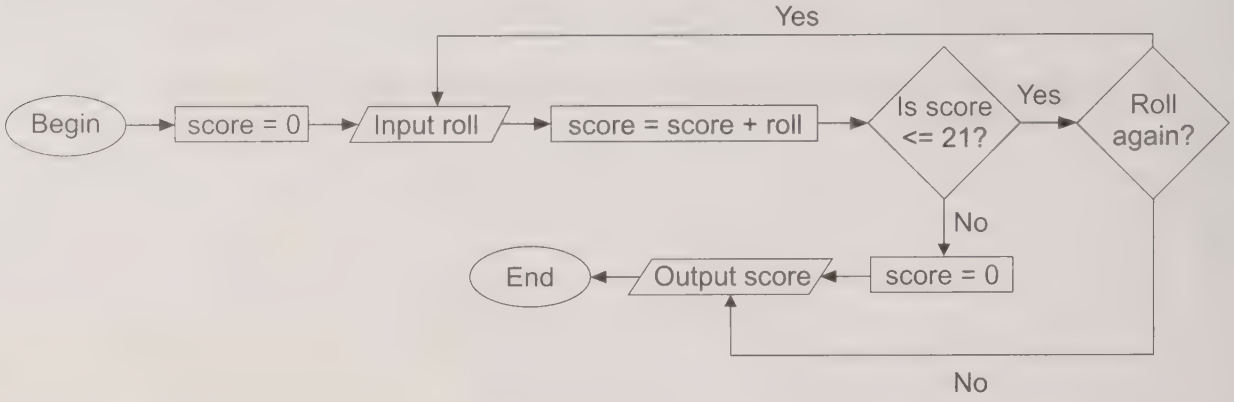
1 .....

2 .....

[2 marks]

7. Tony is writing a program to calculate a player's score in a dice game.
A player starts with a score of 0 and rolls a six-sided dice as many times as they want. After each roll they add the number the dice lands on to their score. The aim is to get as close to 21 as possible. If you go over 21 you get a score of 0.

The flow diagram below shows an initial design for his program.



(a) Tony designs a test plan for his program.
Explain why it's important for the test plan to include erroneous test data.

.....

.....

[2 marks]

(b) Tony tests each of the following sequences of inputs in his program.
For each test, state the type of test data, the intended outcome, and what the actual outcome will be with his program's current design.

(i) 5, 6, 4, 3, 2

Type of data: .....

Intended Outcome: .....

Actual Outcome: .....

[3 marks]

(ii) 5, 9, 3, 2

Type of data: .....

Intended Outcome: .....

Actual Outcome: .....

[3 marks]

(c) Identify **two** ways that Tony could use input validation in his program.

1 .....

2 .....

[2 marks]

8. A comic book shop stores information about each of its comics in records. The table below shows two records stored in the `comics` table.

ID No.	Title	Date published	Length	Genre	Rating
0001	Hike of Hope	04-05-2015	82	Adventure	5
0002	Space Voyage	05-09-2015	65	Science Fiction	4

(a) Explain why the comic book shop has chosen to store this information in records rather than an array.

.....

.....

.....

[2 marks]

(b) Write an SQL query to return:

(i) the titles of all Science Fiction comics.

.....

[2 marks]

(ii) the titles and lengths of all the comics that have fewer than 50 pages and a rating of 3.

.....

[2 marks]

(iii) all the fields for comics with titles that begin with the letter H.

.....

[2 marks]

9. The Prime Koalas are a band consisting of four members: John on guitar, Paul on bass, Cheryl on vocals and Ida on drums.

(a) Complete the code below to generate a 2D array containing the names and instruments of all the band members.

```
array primeKoalas[4, 2]
primeKoalas[0, 0] = "John"
primeKoalas[1, 0] = "Paul"
primeKoalas[2, 0] = "Cheryl"
primeKoalas[3, 0] = "Ida"
```

[2 marks]

(b) John wants to open the empty text file called musicians.txt and write to it.
Each line of the file should contain a different band member and their instrument.
Write an algorithm that uses your array in part (a) to write this information to the text file.

.....

.....

.....

.....

.....

.....

.....

.....

[3 marks]

(c) The sub program `toSpeech()` takes a string as a parameter, turns the string into audio data and reads it out loud. Write a procedure using `toSpeech()` that takes a file name as a parameter and reads out all the data in the file.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[5 marks]

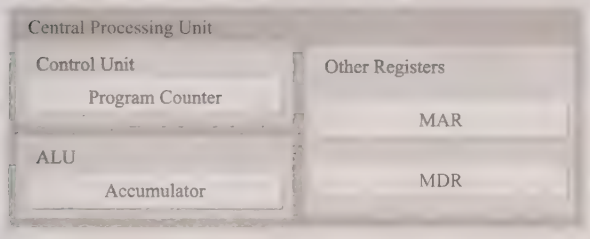
END OF QUESTIONS

Section One — Components of a Computer System

Page 4 (Warm-Up Questions)

1 E.g.
- Power supply
- Case cooling fan
- CPU
- Heat sink
- Optical drive
- RAM
- Hard Disk Drive/HDD
- Graphics card/GPU
- Motherboard

2



3 a) The accumulator stores the results of calculations done by the ALU.

b) The MAR holds any memory address about to be used by the CPU.

c) The MDR holds data or instructions that have been fetched from / are about to be written to memory.

Page 5 (Exam Questions)

3 a) A computer system built into another device. ***[1 mark]***

b) Any **two** devices, e.g.
- Dishwasher ***[1 mark]***
- MP3 player ***[1 mark]***
- Digital thermometer ***[1 mark]***
- Washing machine ***[1 mark]***
- Manufacturing machinery ***[1 mark]***

[2 marks available in total]

c) Any **two** benefits explained, e.g.
- Embedded systems are far smaller than general purpose computers ***[1 mark]*** which means microwaves can be made more compact. ***[1 mark]***
- Embedded systems are cheaper to produce than general purpose computers ***[1 mark]*** which can reduce the costs / sale price of microwaves. ***[1 mark]***
- Embedded systems tend to be more reliable than general purpose computers ***[1 mark]*** which means that microwaves are less likely to break. ***[1 mark]***

[4 marks available in total]

4 a) Any **two** functions, e.g.
- The control unit executes instructions. ***[1 mark]***
- It follows the fetch-decode-execute cycle. ***[1 mark]***
- It controls the flow of data within the CPU. ***[1 mark]***
- It controls the flow of data between the CPU and other parts of the computer system (such as memory, and input and output devices). ***[1 mark]***

[2 marks available in total]

b) E.g. The ALU carries out arithmetic operations, e.g. addition, subtraction and multiplication (using repeated addition). ***[1 mark]*** It performs logic operations on binary data, such as AND, NOT, and OR. ***[1 mark]***

c) E.g. The cache is extremely fast memory in the CPU. ***[1 mark]*** It stores regularly used data or instructions. ***[1 mark]*** The CPU can access data stored in the cache much faster than retrieving it from RAM. ***[1 mark]***

Page 10 (Warm-Up Questions)

1 RAM is volatile memory that can be read from and written to, used to store files and applications while they are in use. ROM is non-volatile memory that can generally only be read, which contains the startup instructions for the computer.

2 A clock speed of 3 GHz means that the single-core processor can process 3 billion instructions per second.

3 a) Magnetic tape

b) E.g. Optical disk / CD / DVD / Blu-Ray™

c) Hard disk drive (HDD)

4 E.g.
- Optical disks tend to have lower capacity.
- USB pen drives have faster read/write speeds.
- Optical disks are easily scratched.
- USB pen drives are more reliable for repeatedly rewriting to.
- USB pen drives are pocket-size so are easier to carry.

Page 11 (Exam Questions)

3 a) E.g.
- Secondary storage is needed to store data and software in the long term. ***[1 mark]***
- Secondary storage is non-volatile memory, so retains data when there is no power. ***[1 mark]***
- Computers could not function without permanent data storage, as all software and data would be lost when switched off. ***[1 mark]***
- Secondary storage has a high capacity, so you can store a lot more data. ***[1 mark]***

[3 marks available in total]

b) Any **two** advantages, e.g.
- Optical discs have a low cost per GB. ***[1 mark]***
- They are highly portable. ***[1 mark]***
- They are durable against shock and water damage. ***[1 mark]***

Any **two** disadvantages, e.g.
- They are very slow to write to. ***[1 mark]***
- They require an optical drive to be read / written. ***[1 mark]***
- They can be scratched easily. ***[1 mark]***
- They have a low capacity compared to other forms of storage, e.g. flash memory cards. ***[1 mark]***

[4 marks available in total]

4 a) Cache is much faster than RAM. ***[1 mark]*** The larger the cache, the more data can be stored for quick access by the CPU, meaning the CPU should perform better. ***[1 mark]***

b) E.g. Jackson's CPU has more cores than Will's CPU, which should mean better performance. ***[1 mark]*** It also has a larger cache than Will's, which should again lead to better CPU performance. ***[1 mark]*** On the other hand Will's CPU has a higher clock speed than Jackson's, so there is a chance that Will's may give better performance than Jackson's. ***[1 mark]*** Overall, it is hard to tell whether Will's CPU will offer better performance, therefore it seems unwise to buy Will's CPU, as it may be no better than Jackson's current one. ***[1 mark]***
If you'd decided that Will's CPU was the best option, you'd still get the marks as long as you'd put together a sensible argument based on comparisons of the CPU specs.

c) E.g. Increasing the amount of RAM increases the amount of data / number of applications that the computer can hold in memory. ***[1 mark]*** Jackson may not use all of the current RAM in his computer, as he may use undemanding software or he may not open many programs at once ***[1 mark]*** so adding more RAM will not improve performance. ***[1 mark]***

[2 marks available in total]

Page 17 (Warm-Up Questions)

1 E.g.
- User account control
- Password/pin protection
- Pattern locking
- Fingerprint/retina scanning

2 A compression utility reduces the size of a file so that it takes up less disk space/transfers quicker/etc.

3

	Open Source	Proprietary
OS	Linux	macOS®
Utility		Windows® Firewall
Other	VLC Media Player	Adobe® Photoshop®

Page 18 (Exam Questions)

3 a) E.g.
- Open source software is software where the source code is made freely available. ***[1 mark]***
- Users can legally modify the source code to make their own version of the software. ***[1 mark]***
- Modified versions of the original source code can be shared under the same license as the original software. ***[1 mark]***

[2 marks available in total]

b) Any **one** advantage, e.g.
- It's usually free of charge. ***[1 mark]*** Free software can be used as a marketing tool — once the device is paid for, the user won't need to pay for any more software. ***[1 mark]***
- The software's source code can be adapted by users ***[1 mark]*** which may increase the functions of the TV-PC. ***[1 mark]***
- Popular open source software can be very reliable as problems may be quickly fixed by the community or developers ***[1 mark]*** meaning Ioteck may not need to work on / push out their own updates. ***[1 mark]***

Any **one** disadvantage, e.g.
- They may have unpatched holes in security ***[1 mark]***, which could worry customers and reduce sales of the product. ***[1 mark]***
- If there are problems with the open source software ***[1 mark]*** Ioteck may not have any support from the software developers. ***[1 mark]***
- Ioteck won't be able to make money by selling their own proprietary software, ***[1 mark]*** so they may make less profit from the product overall. ***[1 mark]***

[4 marks available in total]

4 a) A full backup is where a copy is taken of every file on the system. ***[1 mark]*** An incremental backup is where only the files modified or created since the last backup are copied. ***[1 mark]***

b) E.g. The company could do full backups once a fortnight, ***[1 mark]*** but do incremental backups twice a day. ***[1 mark]*** Every 6 months, the company could use data compression software to compress the previous 6 month's backups in order to reduce the file size. ***[1 mark]*** Backup disks / tapes can then be placed in a fireproof box / locked safe or kept on a different site to protect against fire / flood / theft. ***[1 mark]***

5 E.g.
- When applications / programs / files are opened, the OS moves the necessary parts to memory. ***[1 mark]***
- The OS will remove unneeded data from memory, e.g. when programs or files are closed. ***[1 mark]***
- The OS divides memory into segments. When different programs are used, their data is placed into different segments so that running applications can not write over or interfere with each other. ***[1 mark]***
- The OS organises the movement of data to and from virtual memory. ***[1 mark]***
- The OS divides CPU time between running applications / programs / processes, as it can only process one at a time. ***[1 mark]***
- The OS can prioritise CPU time for different programs in order for them to be processed in the most efficient order. ***[1 mark]***

[6 marks available in total — award a maximum of 4 marks for points about RAM or CPU time]

Section Two — Networks

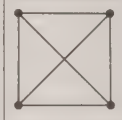
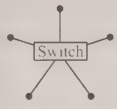
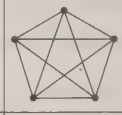
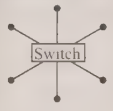
Page 25 (Warm-Up Questions)

1 E.g.
- Network Interface Controller (NIC)
- Switch
- Ethernet cables
- Wireless Access Point (WAP)

2 Clients send requests for data/services to the server, which processes the requests and then replies.

3 E.g. video calling/file sharing/etc.

4

No. of devices	Star	Partial Mesh	Full Mesh
4			
5			
6			

For the partial mesh topology, there are lots of possible answers — all you need is for all nodes to be indirectly linked to each other and for there to be more than one way to get between some of the nodes.

Page 26 (Exam Questions)

2 a) Ethernet is wired, Wi-Fi® is wireless ***[1 mark]***

b) Switch: directs data between devices in a LAN. ***[1 mark]***
Router: transmits data between separate networks / across a WAN / across the Internet. ***[1 mark]***
WAP: allows devices to connect to a LAN wirelessly. ***[1 mark]***

3 Any **one** advantage,
E.g. If a single device or connection fails on a mesh network, the network still functions as data can go along a different route. ***[1 mark]*** In a star network, central switch failure will cause the whole network to fail / a single connection failure will cut that device off from the network. ***[1 mark]***

Any **one** disadvantage,
E.g. Devices in a mesh network need many more connections than in a star network. ***[1 mark]*** This requires a lot of cabling which can be expensive / impractical for networks with a large number of nodes. ***[1 mark]***

4 a) A Peer-to-Peer network is a group of devices connected to share data with each other ***[1 mark]*** in which all devices are equal / connected without a server. ***[1 mark]***
A Client-Server network is a group of devices connected to a central computer (server) ***[1 mark]*** which manages the network / processes requests from devices (clients) / stores files centrally. ***[1 mark]***

b) Any **two** benefits, e.g.
- User files are stored centrally so resources are used more efficiently / files are less likely to be duplicated. ***[1 mark]***
- Software is easier to install / update because it is centrally stored. ***[1 mark]***
- Servers are more reliable than peer machines. ***[1 mark]***
- It is easier to backup data centrally. ***[1 mark]***
- There is greater security as anti-malware software can be installed centrally / user access to files can be controlled. ***[1 mark]***

Any **two** drawbacks, e.g.
- Client-Server networks are expensive to set up. ***[1 mark]***
- The business would need to employ an IT specialist to maintain the server / network. ***[1 mark]***
- If the server fails then all the client computers lose access to their files / client computers are dependent on the server. ***[1 mark]***

[4 marks available in total]

Page 34 (Warm-Up Questions)

1

Protocol	Function
TCP	Sets rules for how devices connect on the network / splits data into packets / reassembles packets into original data / checks data is sent and delivered.
IP	Responsible for packet switching.
HTTP	Used by web browsers to access websites / communicate with web servers.
HTTPS	A more secure version of HTTP.
FTP	Used to access, edit and move files on other devices.
SMTP	Used to send emails / transfer emails between servers.
IMAP	Used to retrieve emails from a server. The user downloads a copy of the email and the server holds the original email until the user deletes it.
POP3	Used to retrieve emails from a server. The server holds the email until the user downloads it, at which point the server deletes it.

2 a) Trojan

b) Rootkit

c) Worm

d) Ransomware

Pages 35-36 (Exam Questions)

2 a) The Internet is a global network of networks. ***[1 mark]*** The World Wide Web is a collection of websites hosted on web servers. ***[1 mark]***

b) A Domain Name Server translates a website's domain name into its IP address / stores domain names of websites in a directory. ***[1 mark]***

3 a) E.g.
- IP addresses are used to identify devices on a network / on the Internet. ***[1 mark]***
- Routers use IP addresses to direct data packets to the correct destination. ***[1 mark]***

[2 marks available in total]

b) E.g.
- Holly's laptop checks periodically that all packets have been received. ***[1 mark]***
- If Holly's laptop does not receive all the packets in a certain amount of time it sends a timeout message back to the smartphone. ***[1 mark]***
- Mahindar's smartphone resends the email / packet. ***[1 mark]***
- Holly's laptop uses the packet numbers to reassemble them into the correct order. ***[1 mark]***

[3 marks available in total]

4 a) E.g.
- Different user access levels prevent students from accessing the same data as teachers, including sensitive data like their peers' personal information. ***[1 mark]***
- Different user access levels prevent a student from maliciously deleting or editing data. ***[1 mark]***
- Different user access levels prevent students accidentally deleting or editing important files. ***[1 mark]***
- Different user access levels allow network administrators to flexibly change the amount of access students and staff have to certain files. ***[1 mark]***

[3 marks available in total]

b) i) A brute force attack is a type of attack on a network which uses trial and error to crack passwords ***[1 mark]*** by employing automated software to produce hundreds of likely combinations. ***[1 mark]***

ii) Any **two** measures, e.g.
- The school can lock access to user accounts after a certain number of password attempts. ***[1 mark]***
- The school can ensure that strong passwords are used / the school can ensure that passwords are long enough / the school can ensure that passwords are made of a mix of different types of character. ***[1 mark]***
- The school can add additional security checks, like using a secret question or CAPTCHA test. ***[1 mark]***

[2 marks available in total]

5 a) E.g.
- A MAC address is a unique identifier. ***[1 mark]***
- A MAC address is assigned to the hardware of every network-enabled device. ***[1 mark]***
- MAC addresses are used to direct data to the right device on a network. ***[1 mark]***

[2 marks available in total]

b) i) A layer of network protocols is a group of protocols with similar functions ***[1 mark]*** that cover one particular aspect of network communications. ***[1 mark]***

ii) E.g.
- Data link layer ***[1 mark]*** Function: passing data over the physical network / sending data bits as electrical signals over cables, wireless and other hardware. ***[1 mark]***
- Network layer ***[1 mark]*** Function: making connections between networks / directing data packets / handling traffic. ***[1 mark]***
- Transport layer ***[1 mark]*** Function: controlling data flow / splitting data into packets / checking packets are correctly sent and delivered. ***[1 mark]***
- Application layer ***[1 mark]*** Function: turning data into websites and other applications / turning websites and other applications into data. ***[1 mark]***

[2 marks available in total]

iii) Any **three** benefits, e.g.
- Layers break network communication into manageable pieces. ***[1 mark]***
- Layers allow developers to focus on one area of the network without worrying about the others. ***[1 mark]***
- Layers are self-contained. ***[1 mark]***
- There are set rules for each layer. ***[1 mark]***
- Layers allow interoperability / layers make companies produce compatible, universal hardware and software. ***[1 mark]***

[3 marks available in total]

6 Points you might include:

Advantages of the cloud

- Users can access files from any location, so the company and authors can work on the same files without having to email or post manuscripts.
- Cloud storage is managed by the hosting company which will be a cheaper alternative to managing their own storage.
- The hosting company manages the security of data in the cloud, so the publishing company does not need to spend time securing its data.
- The hosting company is responsible for backing up data in the cloud, so the publishing company does not need to invest in any additional hardware to ensure data is backed up correctly.
- The publishing company could use cloud-based software rather than installing it on their machines and keeping it up to date. This could give writers access to the same software.

Disadvantages of the cloud

- An Internet connection is required to access the cloud, and maintaining a steady Internet connection can be difficult in rural areas.
- The publishing company is dependent on the hosting company for the security in the cloud, meaning the publishing company has very little control over the security of its data.
- The publishing company is dependent on the hosting company for backing up their data.
- Cloud software may require a monthly subscription which may be more expensive than buying computer licences.

How to mark your answer:

- Two or three brief points with very little explanation. ***[1-2 marks]***
- Three or four detailed points covering both advantages and disadvantages. ***[3-4 marks]***
- Five or more detailed points that form a well-written, balanced discussion, covering both advantages and disadvantages. ***[5-6 marks]***

Make sure your answer is relevant to the situation you're given — the company in the question has particular needs and qualities which you shouldn't ignore.

7 Points you might include:

The threats posed to the firm's network

- Hackers could use rootkits, spyware and other malware to steal confidential information.
- Employees unaware of the potential dangers could be tricked into giving criminals sensitive information through social engineering.
- Disgruntled employees could use their position to attack the network, e.g. by releasing malware onto the network from a USB drive.
- Hackers with packet sniffers or other similar tools could intercept and read information entering or leaving the company's network.
- Hackers could use a brute force attack to crack weak passwords.

What a good network policy would include

- Automatic encryption of all data leaving and entering the network.
- Installing anti-malware and firewall software.
- Regular penetration testing to find problems in the network security.
- Education of employees on the dangers of social engineering.
- Mandatory use of strong passwords / passwords that are changed regularly.
- An acceptable use policy that all employees must sign.
- Controlling physical access to hardware / the network, e.g. keeping servers in locked rooms.
- Different user access levels given to different groups of users.

How a good network policy would prevent potential attacks on the firm's network.

- A firewall would prevent harmful malware from entering the network.
- Education of employees could prevent social engineering.
- Different user access levels and physical security measures could limit the dangers of an insider attack.
- Encrypting data could prevent intercepted data from being read by hackers and criminals.
- Using strong passwords could prevent successful brute force attacks.

How to mark your answer:

- Two or three points with very little explanation. ***[1-2 marks]***
- Three to five points with detailed explanation. ***[3-5 marks]***
- Six or more detailed points that form a well-written, balanced discussion. ***[6-8 marks]***

Section Three — Issues

Page 43 (Warm-Up Questions)

1

	Censorship	Surveillance
A business monitors what their employees view online.		✓
A country's government blocks access to Facebook®.	✓	
A government agency intercepts emails containing certain words.		✓
A school restricts access to harmful websites.	✓	
An Internet Service Provider collects data on browsing habits.		✓

2 E.g.

- Eyestrain — use suitable lighting / keep a good distance from the screen / take regular breaks.
- Repetitive Strain Injury (RSI) — ensure good posture / arrange your desk appropriately / take regular breaks.
- Back pain — ensure good posture / use adjustable equipment / sit at a suitable angle.

3 The digital divide is the separation between people who have ready access to technology and those who don't, who tend to be at a disadvantage because of this.

Page 44 (Exam Questions)

2 a) An individual or group of people with an interest in a particular decision / are affected by a particular outcome. ***[1 mark]***

b) Any **two** stakeholders, e.g.

- Business owners / managers ***[1 mark]*** — positive ***[1 mark]*** because they receive increased profits and do not have to pay the staff. ***[1 mark]***
- Checkout staff ***[1 mark]*** — negative ***[1 mark]*** because they have lost their jobs. ***[1 mark]***

3 a) E.g.

- The Internet / social media / email mean Tom can be contacted at any time of day. ***[1 mark]***
- Tom may be expected to carry a smartphone so he can be contacted by his boss at all times. ***[1 mark]***
- Tom's smartphone may alert him when he receives work emails from clients. These can be hard to ignore. ***[1 mark]***

[2 marks available in total]

b) E.g.

- Companies try to influence people into using their new product, e.g. by using advertisements. ***[1 mark]***
- Many children may feel peer pressure to buy the new devices for fear of being bullied by their classmates. ***[1 mark]***
- Parents can feel pressured to buy the latest technology for their children. ***[1 mark]***

[2 marks available in total]

c) Any **one** reason, e.g.
- They might not have a lot of money ***[1 mark]*** so cannot afford to buy expensive electronic devices. ***[1 mark]***
- They might live in a rural area ***[1 mark]*** so could have poor network coverage. ***[1 mark]***
- They might have little knowledge of how to use the Internet or electronic devices ***[1 mark]*** so feel too intimidated to use them. ***[1 mark]***

[2 marks available in total]

4 Points you might include:

Stakeholders
- Manufacturing businesses can cut costs by getting the same products made without having to pay any wages.
- Workers in the manufacturing sector could become unemployed, as robots take over their jobs.
- Consumers could be able to buy the same products for less, as the costs involved in making them could be lower.

Technology
- The hardware and software of robots may not be sophisticated enough to fully replicate the work of human employees.
- The increased use of robots in the workplace could help improve them as problems can be identified and fixed.
- Successful use of robots in manufacturing could lead to their application in other areas of work.

Ethical issues
- Manufacturing businesses could leave hundreds of people without jobs in order to pursue profit.
- There is currently a lack of awareness and rules around the use of robots in the workplace.
- By allowing robots to do the routine jobs, workers are free to do more interesting, creative and fulfilling work.
- New jobs are created to program, maintain and manufacture the robots.
- Robots can perform hazardous tasks, meaning there could be fewer injuries in the workplace.

How to mark your answer:
- Two or three brief points with very little explanation. ***[1-2 marks]***
- Three or four detailed points covering at least two of: stakeholders, technology and ethical issues. ***[3-4 marks]***
- Five or more detailed points that form a well-written, balanced discussion, covering all of: stakeholders, technology and ethical issues. ***[5-6 marks]***

Page 48 (Warm-Up Questions)

1 Copper, platinum, plastic, mercury, gold and silver.

2 a) Data Protection Act 1998

b) Computer Misuse Act 1990

c) Freedom of Information Act 2000

d) Data Protection Act 1998

e) Computer Misuse Act 1990

3 a) The copyright holder must be given credit.

b) The work cannot be used for financial or commercial gain.

c) Modified works can only be distributed with the same terms as those applied to the original.

d) The work cannot be modified or built upon.

Page 49 (Exam Questions)

3 a) A way of protecting intellectual property / a way of protecting something that has been created / a way of protecting written and recorded content, e.g. books, music, films, software and video games. ***[1 mark]***

b) Works in the public domain are those whose copyright has expired / works which do not have any copyright attached to them ***[1 mark]*** meaning they can be shared and copied without normal copyright rules being applied. ***[1 mark]***

4 a) E-waste is discarded computer material / discarded electronic material. ***[1 mark]***

b) Any **two** reasons, e.g.
- Smartphones are portable, so can be broken easily, for example by dropping them. ***[1 mark]***
- Smartphones are not built to last more than a few years. ***[1 mark]***
- It is often cheaper to replace a smartphone than it is to repair it. ***[1 mark]***
- Smartphone manufacturers release new devices on a regular basis and use advertisements to influence people into buying them. ***[1 mark]***
- People often want to buy the newest technology and are happy to discard their old smartphone regularly. ***[1 mark]***
- Retailers only provide short warranties on many smartphones. ***[1 mark]***
- Many people feel pressured by their peers to upgrade to the latest smartphone. ***[1 mark]***

[2 marks available in total]

c) E.g.
- E-waste is sent to landfill sites ***[1 mark]*** where toxic chemicals can enter groundwater / harm wildlife. ***[1 mark]***
- The short life span of devices also means more natural resources have to be extracted to make new devices ***[1 mark]*** which causes pollution and depletes scarce resources. ***[1 mark]***
- Manufacturing devices uses up a lot of electricity ***[1 mark]*** which is often generated using non-renewable resources. ***[1 mark]***

[4 marks available in total]

d) E.g.
- People can take devices to local collection facilities to be correctly disposed of according to WEEE regulations ***[1 mark]*** rather than throwing them out with general waste. ***[1 mark]***
- The government can put pressure on companies or local authorities to ensure WEEE regulations are being followed. ***[1 mark]*** The government can set recycling targets to increase the amount of e-waste that is recycled. ***[1 mark]***
- Old devices can be refurbished and reused ***[1 mark]*** and their raw materials can be recycled. ***[1 mark]***

[2 marks available in total]

5 Points you might include:
- The cinema should ask permission from the customer before storing their data.
- The cinema should ask the customer to sign an agreement as to how the information should be used.
- The cinema should only use the data to make it easier for customers to book seats and for contacting them for details on future films.
- The cinema should give customers the option to unsubscribe from their service, so that their data is removed from the cinema's computer system if they want.
- The cinema should be active in keeping its data up to date and allow their customers to change their details.
- The cinema should not give customers' data to third party organisations unless the customer permits it.
- The cinema should ensure that customers' data is held safely and securely, e.g. by using a firewall on its computer system and encrypting the data.

How to mark your answer:
- Two or three brief points with very little explanation. ***[1-2 marks]***
- Three or four detailed points that show a good understanding of the Data Protection Act. ***[3-4 marks]***
- Five or more detailed points that show a good understanding of the Data Protection Act and clearly apply it to the situation. ***[5-6 marks]***

Think about each principle of the Data Protection Act and apply it to this situation.

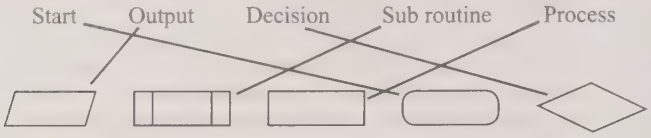
Section Four — Algorithms

Page 54 (Warm-Up Questions)

1 Decomposition, Abstraction and Algorithmic Thinking

2 An algorithm is a process or set of instructions used to solve a problem or carry out a task.

3 B and D

4 

Page 55 (Exam Questions)

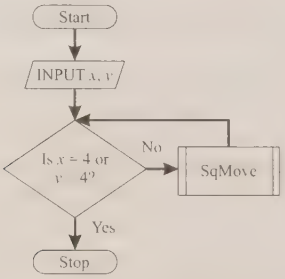
2 a) It asks the user to input a height and width. ***[1 mark]***
It then multiplies these values together ***[1 mark]***
to get the area and prints the value of the area. ***[1 mark]***

b) $5 \times 10 = 50$ ***[1 mark]***

3 a) E.g. Abstraction is picking out important details and ignoring irrelevant ones. The file uploading service will focus on the important details like the file name and ignore the unimportant details like the contents of each file.
[3 marks available — 1 mark for a definition of abstraction, 1 mark for an example of a detail to ignore, 1 mark for an example of a detail to focus on]

b) E.g. Decomposition breaks the programming task down into smaller problems. A programmer might focus on 'How will the service keep track of files already uploaded?' or 'How will the service compare file names?' and try to solve each programming problem individually.
[3 marks available — 1 mark for a definition of decomposition, 1 mark for each example of decomposition up to a maximum of 2 marks]

4 Using Start / Begin. ***[1 mark]***
Asking user to input x, y. ***[1 mark]***
Using SqMove (with correct subroutine box). ***[1 mark]***
Decision box with appropriate question. ***[1 mark]***
Creating a loop to repeat SqMove. ***[1 mark]***
Using Stop / End. ***[1 mark]***
E.g.



Page 60 (Warm-Up Questions)

1 a) Middle item = (7 + 1) / 2 = 4th item = 11
11 is bigger than 8 so take left hand side.
3, 6, 8
Middle item = (3 + 1) / 2 = 2nd item = 6
6 is smaller than 8 so take right hand side.
8
Middle item = (1 + 1) / 2 = 1st item = 8
Stop searching as 8 has been found.

b) Check 1st item: $3 \neq 11$.
Check 2nd item: $6 \neq 11$.
Check 3rd item: $8 \neq 11$.
Check 4th item: $11 = 11$.
Stop searching as 11 has been found.

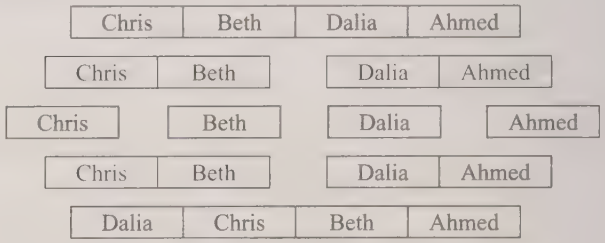
2 See page 57.

3 a) *1st pass (2 swaps)*:
Chris Beth Dalia Ahmed
Beth Chris Dalia Ahmed
Beth Chris Ahmed Dalia

2nd pass (1 swap):
Beth Chris Ahmed Dalia
Beth Ahmed Chris Dalia

3rd pass (1 swap):
Beth Ahmed Chris Dalia
Ahmed Beth Chris Dalia

4th pass (no swaps — list is in order):
Ahmed Beth Chris Dalia
You don't necessarily need to do the last pass — since there are four items in the list, the algorithm will only take at most three passes to get the list in order.

b) 

c)

Chris	Beth	Dalia	Ahmed
Beth	Chris	Dalia	Ahmed
Beth	Chris	Dalia	Ahmed
Ahmed	Beth	Chris	Dalia

4 Merge Sort

Page 61 (Exam Questions)

3 *1st pass (3 swaps)*:
5.32 5.50 5.39 6.50 6.28 6.14
5.32 5.39 5.50 6.50 6.28 6.14
5.32 5.39 5.50 6.28 6.50 6.14
5.32 5.39 5.50 6.28 6.14 6.50

2nd pass (1 swap):
5.32 5.39 5.50 6.28 6.14 6.50
5.32 5.39 5.50 6.14 6.28 6.50

3rd pass (no swaps — list is in order):
5.32 5.39 5.50 6.14 6.28 6.50
[4 marks available — 1 mark for each correct swap, 1 mark for correct ordered list]

4 a) Compare butterscotch to mint. ***[1 mark]***
Mint is greater so split and take the left side. ***[1 mark]***
A further comparison. ***[1 mark]***
Correct identification of butterscotch. ***[1 mark]***
E.g.
Middle item = (5 + 1) / 2 = 3rd item = mint.
Compare mint with butterscotch.
Butterscotch comes before mint, so take left hand side.
The list is: Butterscotch, Chocolate.
Middle item = (2 + 1) / 2 = 1.5 = 2nd item = chocolate.
Compare chocolate to butterscotch.
Butterscotch comes before chocolate, so take left hand side.
Middle item = (1 + 1) / 2 = 1st item = butterscotch.
Stop searching as butterscotch has been found.

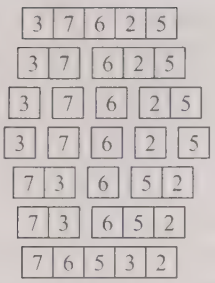
b) It is much more efficient / takes fewer steps for large lists of items. ***[1 mark]***
You won't get the mark for just saying it's quicker or more efficient.

5 a) i)

3	**7**	6	2	5
7	3	**6**	2	5
7	6	3	**2**	5
7	6	3	2	**5**
7	6	5	3	2

[4 marks available — 1 mark for each row from rows 2-5]

ii) E.g.



[4 marks available — 1 mark for correctly splitting the list into single items, 1 mark for each correct merging row]

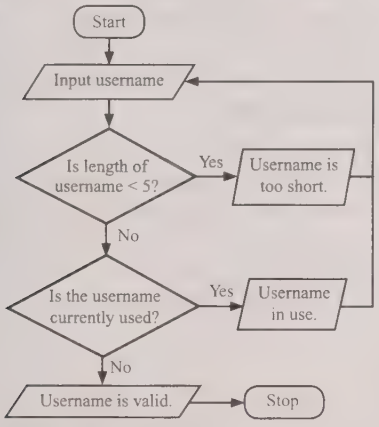
The list doesn't split evenly so there is more than one way to get the right answer depending on how you split the items.

b) Any **one** benefit of insertion sort contrasted to merge sort, e.g.
- Insertion sort uses less memory ***[1 mark]*** as, unlike a merge sort, all the sorting is done on the original list. ***[1 mark]***
- Insertion sort is quicker to check if a list is in order ***[1 mark]*** as a merge sort would still need to go through the splitting–merging process. ***[1 mark]***

[2 marks available in total]

Page 62 (Revision Questions)

8 E.g.



11 a) Binary Search:
The middle item is the 4th item, *Dagenham*, which comes before *Morpeth*. So lose first half of list to leave:
Morpeth, Usk, Watford
The middle item of the new list is the 2nd item, *Usk*, which comes after *Morpeth* so lose second half of list to leave:
Morpeth
The middle item of the new list is the 1st item, which is *Morpeth*, so you've found the correct item.

Even when there is one entry left you still have to carry on with the algorithm to check that it is the correct entry.

b) Linear Search:
Ashington ≠ *Morpeth*
Brecon ≠ *Morpeth*
Chester ≠ *Morpeth*
Dagenham ≠ *Morpeth*
Morpeth = *Morpeth*
You've found the correct item.

13 b) *1st pass (4 swaps)*:
O, B, A, P, G, L
B, O, A, P, G, L
B, A, O, P, G, L
B, A, O, G, P, L
B, A, O, G, L, P

2nd pass (3 swaps):
B, A, O, G, L, P
A, B, O, G, L, P
A, B, G, O, L, P
A, B, G, L, O, P

3rd pass (no swaps — list is in order):
A, B, G, L, O, P

16 a)

b) 8 7 5 1 3 6 4 2
7 8 5 1 3 6 4 2
5 7 8 1 3 6 4 2
1 5 7 8 3 6 4 2
1 3 5 7 8 6 4 2
1 3 5 6 7 8 4 2
1 3 4 5 6 7 8 2
1 2 3 4 5 6 7 8

Section Five — Programming

Page 68 (Warm-Up Questions)

1 a) String b) Integer c) Real/Float d) Boolean

2 a) 40 b) 1 c) 4 d) 19

3 a) True b) True c) False d) True

4 E.g. The value of a constant is set at design time and does not change as the program is running. The value of a variable might change as the program is running.

5 a) 7 b) lob c) LOBSTER d) bst

Page 69 (Exam Questions)

3 a) Boolean ***[1 mark]*** — the variable can only take two values, either pressed or not pressed, i.e. true or false. ***[1 mark]***

b) Integer ***[1 mark]*** — it's measuring the number of whole seconds and whole numbers are best stored as integers. ***[1 mark]***

4 Investment is declared as a constant but its value changes / it is not assigned a value when it is declared. ***[1 mark]***
Interest is declared as an integer but it should be able to take non-integer values. ***[1 mark]***

5 a) Joining together two or more strings. ***[1 mark]***

b) ORA2000 ***[1 mark]***

c) 05 fruit = fruit.upper ***[1 mark]***
06 prodID = fruit.subString(0, 3) ***[1 mark]***
+ str(volume) ***[1 mark]***

Page 74 (Warm-Up Questions)

1 a) Iteration b) Selection c) Selection
d) Iteration e) Selection f) Iteration

2 E.g. A SWITCH-CASE statement makes selections based solely on the value of one variable, an if-elseif statement makes selections based on any conditions that are true or false.

3 a) Count-controlled loop: FOR loop
Condition-controlled loop: E.g. DO UNTIL loop

b) A FOR loop will iterate a specified number of times.
A DO UNTIL loop will keep iterating until a specific condition is satisfied.

4 a) False b) True c) True d) True

Page 75 (Exam Questions)

3 a) Sequence ***[1 mark]***

b) 2 hours and 30 minutes ***[1 mark]***

4 a) Using an IF-ELSE or nested IF statement. ***[1 mark]***
Using the correct conditions to check all settings. ***[1 mark]***
Changing the temperature correctly for each setting. ***[1 mark]***

```
E.g.
int setting, temperature
if setting = = 3 then
  temperature = 50
elseif setting = = 2 then
  temperature = 30
elseif setting = = 1 then
  temperature = 20
else
  temperature = 0
endif
```

b) Any **two** reasons, e.g.
- You only need to check the value of one variable. ***[1 mark]***
- The setting variable only has a set number of possible values. ***[1 mark]***

5 Using an appropriate selection statement. ***[1 mark]***
A Boolean condition that checks each of the conditions. ***[1 mark]***
Allowing the dryer to start if conditions are met. ***[1 mark]***

```
E.g.
real weight
bool allowStart, doorClosed
if (weight > 1.5 AND weight < 15.0) AND doorClosed == true
then
  allowStart = true
else
  allowStart = false
endif
```

6 Count controlled loop to allow 10 games. ***[1 mark]***
Asking for an input of the winner's name for each game. ***[1 mark]***
A selection statement for the winner of each game. ***[1 mark]***
Adding 1 to the winner's score. ***[1 mark]***
A selection statement to find the overall winner. ***[1 mark]***
Printing the correct message depending on the scores. ***[1 mark]***

```
E.g.
int karlWin = 0
int johnWin = 0
string winner
for i = 1 to 10
  winner = input("Enter the winner's name")
  switch winner:
    case "Karl":
        karlWin = karlWin + 1
    case "John":
        johnWin = johnWin + 1
  endswitch
next i
if karlWin > johnWin then
  print("The winner is Karl.")
elseif johnWin > karlWin then
  print("The winner is John.")
else
  print("The game is a draw.")
endif
```

To make your algorithm more robust you could have used input validation to make sure winner was either "Karl" or "John". You could also have named the variables differently so that the game could be played by any two players regardless of their name.

Page 83 (Warm-Up Questions)

1 a) Assigns the 4th element in the array to the variable 'player'.

b) Replaces the 6th element of the array with the string "Pele".

2 a) openRead() b) openWrite() c) close() d) readLine()

3 E.g.
- A record can store different data types.
- Names can be given to the different fields.
- Record structures cannot accidently be changed later on.

4 a) SELECT — the fields you want to return.

b) FROM — the table you want to get the data from.

c) WHERE — A condition that has to be true for a record to be returned.

d) LIKE — A pattern of data to look for.

5 Any **three** benefits, e.g.
- You only have to write them once so you don't have to repeat blocks of code.
- You can call them from anywhere in the program.
- You only have to debug them once.
- They will improve the readability / maintainability of the code.
- They break the program down into smaller more manageable chunks.

6 Arguments are the actual values that parameters take when the sub program is called.

Pages 84-85 (Exam Questions)

2 A function that takes a single parameter. ***[1 mark]***
Finding the cube and square of the parameter. ***[1 mark]***
Returning the difference between the cube and square. ***[1 mark]***

```
E.g.
function cubeSquare(numberInt)
  return(numberInt^3 – numberInt^2)
endfunction
```

3 a) String ***[1 mark]***

b) Any **three** reasons, e.g.
- Multiple items of data need to be stored. ***[1 mark]***
- All the data being stored has the same data type. ***[1 mark]***
- The data is split by two categories / can be represented in a table so a 2D array is useful for storing it. ***[1 mark]***
- Stores the data together under one variable name. ***[1 mark]***
- Accessing the information is more efficient. A single command, e.g. *sportsDay[position, event]* can be used to access any name from the array. ***[1 mark]***

[3 marks available in total — at least one reason must specifically mention 2D arrays.]

4 a) 5 ***[1 mark]***

b) A record is a data structure used to store multiple pieces of data about one thing together (e.g. information about a particular car). ***[1 mark]*** A field is one of the items in a record that contains a particular piece of data (e.g. car registration or car make). ***[1 mark]***
Just saying that records are rows of the table and fields are columns will not be awarded any marks.

c)

Make	Type
Stanton	Hatchback
Stanton	Saloon

[2 marks available — 1 mark for each correct record]

5 a) *print(distanceRun[4, 3])* ***[1 mark]***

b) Asking the user to input the runner number. ***[1 mark]***
Using a FOR loop. ***[1 mark]***
Adding all elements correctly. ***[1 mark]***
Printing the total distance. ***[1 mark]***
E.g.
```
int totalDistance = 0
int runner
runner = input("Choose a runner number from 0-3")
for i = 0 to 6
  totalDistance = totalDistance + distanceRun[i, runner]
next i
print(totalDistance)
```

c) A FOR loop going from 0 to 3. ***[1 mark]***
A FOR loop going from 0 to 6. ***[1 mark]***
Using milesConvert() on each element of the array. ***[1 mark]***
E.g.
```
for i = 0 to 3
  for j = 0 to 6
    distanceRun[i, j] = milesConvert(distanceRun[i, j])
  next j
next i
```

6 Opening the story in read mode. ***[1 mark]***
A condition-controlled loop to stop at the end of the file. ***[1 mark]***
Waiting for a keypress input. ***[1 mark]***
Using a selection statement to check the user's input. ***[1 mark]***
Printing the next line of the story. ***[1 mark]***
E.g.
```
char keypress
story = openRead("adventure.txt")
while NOT story.endOfFile()
  input keypress
  if keypress == "y" then
    print(story.readLine())
  endif
endwhile
story.close()
```

7 a) Any **two** differences, e.g.
- Local variables can only be changed and accessed from within the part of the program they're declared in. ***[1 mark]*** Global variables can be changed and accessed from anywhere in the program. ***[1 mark]***
- The same local variable name can be used in different sub programs and declared differently each time. ***[1 mark]*** Global variable names can only be declared once. ***[1 mark]***
- Local variables are declared inside part of a program (e.g. in a sub program). ***[1 mark]*** Global variables are usually declared at the start of the main program. ***[1 mark]***

[4 marks available in total]

b) A function that takes the number of sides the dice have as a parameter. ***[1 mark]***
Using a condition controlled loop. ***[1 mark]***
Simulating two dice rolls. ***[1 mark]***
Increasing the score by 1 after each roll. ***[1 mark]***
Returning the score. ***[1 mark]***
E.g.
```
function rollDouble(side)
  int score = 0
  int x, y
  do
    x = roll(side)
    y = roll(side)
    score = score + 1
  until x == y
  return(score)
endfunction
```

Page 86 (Revision Questions)

9
```
string first_day
first_day = input("Enter the first day of the month")
if first_day == "Sunday" OR first_day == "Monday" then
  print("The month has 5 Mondays")
else
  print("The month has 4 Mondays")
endif
```

11 a) *print(chars[4])*

b)
```
chars[2] = "D"
print(chars)
```

c)
```
for i = 0 to 9
  chars[i] = "N"
next i
print(chars)
```

12
```
array multiply[10, 10]
for i = 0 to 9
  for j = 0 to 9
    multiply[i, j] = i * j
  next j
next i
```

15 The query will return every field of the world_records table.
A record will be returned if the sport is athletics and the surname begins with an "M".

Section Six — Design, Testing and IDEs

Page 94 (Warm-Up Questions)

1 Validation checks make sure that data meets certain criteria before passing it into the program, while sanitisation is used to remove unwanted characters from data before passing it through the program.

2 a) Presence check

b) Look-up table

3 E.g.
- Clear comments
- Indentation
- Properly named variables/subprograms
- Use of local (rather than global) variables where possible

4 a) False b) True c) True d) False

5 Iterative testing puts the program through multiple development cycles, adjusting the requirements after each cycle to make the software closer to what the customer really wants.

6 A compiler translates all of a program's source code into one executable file of machine code.

7 Code Editor: used for writing code.
Breakpoints: a debugging tool that stops the program at certain points.
Error Diagnostics: highlight errors in the program.

Pages 95-96 (Exam Questions)

2 A translator turns the source code into machine code ***[1 mark]*** allowing Cynthia to run the application. ***[1 mark]***

Error Diagnostics highlight errors in the code ***[1 mark]*** allowing Cynthia to easily find and fix errors in her application. ***[1 mark]***

A Code editor allows the user to enter code and includes features like indenting, auto-correct, line numbering and colour coding ***[1 mark]*** making it a lot easier for Cynthia to write and maintain her application. ***[1 mark]***

3 a) E.g. a test plan should take the user down all possible paths of the program. ***[1 mark]*** It should use normal, extreme and erroneous test data. ***[1 mark]*** If any of the tests get a result not equal to the expected outcome then the user knows there is a logic error. ***[1 mark]***

b)

Test Data	Expected Outcome	Reasons for test
Group_Size = 4	210	Check program with data user is likely to input.
Group_Size = 9	460	Check program works with values on the limit.
Group_Size = 12	Display an error message	Check what happens if input is too large.

[5 marks available — 1 mark per box]

4 a) Any **two** reasons, e.g.
- They may need to have a greater control over the program in order to make a program with lower memory use. ***[1 mark]***
- They may need to have a greater control over what the CPU does in order to make a program run quicker. ***[1 mark]***
- They may be trying to maintain old code or hardware. ***[1 mark]***

[2 marks available in total]

b) Assembly languages are more readable for humans than machine code ***[1 mark]*** so can be programmed/edited more easily. ***[1 mark]***

c) Assembler ***[1 mark]***

5 a) Error: Missing bracket on line 4. ***[1 mark]***
Correction: *print("Valid pincode")* ***[1 mark]***

b) Error: Wrong Boolean operator in line 3. ***[1 mark]***
Correction:
if pincode.length >= 4 AND pincode.length <= 6 then
[1 mark]
Tiffany's code currently allows any pincode length. Changing OR to AND makes sure only pincodes with lengths from 4 to 6 characters are allowed.

6 E.g.
- Input validation could make sure that only appropriate data was entered. ***[1 mark]*** E.g. only allowing numbers to be entered for the card number. ***[1 mark]***
- Input validation would ensure the input data matches certain criteria or is of a certain format. ***[1 mark]*** E.g. it can make sure that the expiration month is between 1 and 12. ***[1 mark]***
- Input validation alone will not prevent all errors. ***[1 mark]*** E.g. it cannot check that a customer's card number matches their name. ***[1 mark]***
- Using input sanitisation alongside validation would improve the defensive design ***[1 mark]*** as you'd be able to remove unwanted parts of an input before validating it. ***[1 mark]***

[6 marks available in total]

7 E.g.
- The code editor might include an auto-indentation feature ***[1 mark]*** which will make the code clearer to read. ***[1 mark]***
- The code editor might include an auto-colour feature ***[1 mark]*** which makes it easier to distinguish between different features of the code, e.g. variables and comments. ***[1 mark]***
- Finding logic errors requires a lot of work by the user — the IDE has debugging tools to help ***[1 mark]*** but most of the work is done through testing. ***[1 mark]***
- The IDE won't be able to tell if clear variable names and comments have been used ***[1 mark]*** so it won't be able to help with this, it's up to the developer. ***[1 mark]***
- The IDE can't check if programmers have used a consistent approach in different parts of the program ***[1 mark]*** so it will be up to the developers to maintain good practice. ***[1 mark]***

[6 marks available in total]

Page 97 (Revision Questions)

4 Input validation — the program checks if the user has entered a valid year before breaking out of the DO UNTIL loop.

10 The code doesn't do anything if the age is exactly 16.
Either change the first condition to "*x >= 16*"
or the second condition to "*x <= 16*".

On line 3 of the code, the variable x should be cast as a string before being concatenated with the other strings.
Instead of "*x*", it should read "*str(x)*".

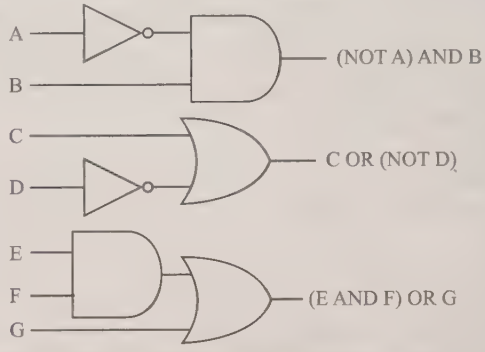
15 E.g.

Test Data	Reason for Testing	Expected Outcome
"tra"	Normal usage of the program.	"art", "rat", "tar"
No input	No input is entered.	Prompt to enter something.
"stm"	Input can't be made into a word.	Returns no words.
"AtN"	Input contains upper and lower case letters.	"tan", "ant", "nat"
"2t1?"	Input contains numbers and symbols.	Error: Unknown character.

Section Seven — Data Representation

Page 101 (Warm-Up Questions)

1 a)



b)

A	B	NOT A	(NOT A) AND B
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	0

C	D	NOT D	C OR (NOT D)
0	0	1	1
0	1	0	0
1	0	1	1
1	1	0	1

E	F	G	E AND F	(E AND F) OR G
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

2 Nibble, Byte, Kilobyte, Megabyte, Gigabyte

3 a) String A: 01101010
String B: 01001011
String C: 0110100101001011

b) String A: 01101011
String B: 01001010
String C: 0110100101001010

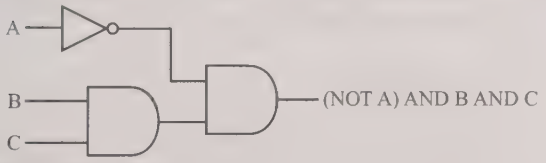
Page 102 (Exam Questions)

3 a)

A	B	C	NOT A	B AND C	(NOT A) AND (B AND C)
0	0	0	1	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	0

[3 marks available — 1 mark for each correct column]

b)



[3 marks available — 1 mark for input A going into a NOT gate, 1 mark for inputs B and C going into an AND gate, 1 mark for outputs going into an AND gate with one output]

4 a) A digit added to the end of a string of numbers / data to check that data has been entered and received correctly. ***[1 mark]*** The value is calculated from the digits in the string. ***[1 mark]***

b) Computers are made up of logic circuits ***[1 mark]*** which use 1 and 0 to show high and low voltage. ***[1 mark]***

c) String 1 contains an error ***[1 mark]*** because it has an odd number of 1s. ***[1 mark]***

d) In the case where an even number of bits are read incorrectly ***[1 mark]*** a parity bit will not detect an error. ***[1 mark]***

Page 108 (Warm-Up Questions)

1 a) 89 b) 1000101

2 11110010

3 Overflow errors occur when binary operations produce results that have more bits than the CPU is expecting to store/process.

4 00010111

5 Multiplying the number by 2.

6 a) 30 b) 198 c) A2 d) 111101

Page 109 (Exam Questions)

3 a) 11010000 ***[1 mark]***

b) A 2 place right shift ***[1 mark]*** gives 00110101. ***[1 mark]***

c) He is not correct.
E.g. 0001 + 0001 = 0010 is a 1 place left shift.
[2 marks available — 1 mark for not correct, 1 mark for a valid explanation]

4 a) Split bytes into nibbles and convert to hexadecimal.
0100 = 4, 0011 = 3, so 01000011 = C = 43. ***[1 mark]***
0100 = 4, 0001 = 1, so 01000001 = A = 41. ***[1 mark]***
0101 = 5, 0100 = 4, so 01010100 = T = 54. ***[1 mark]***

b) i) D = 44 = 01000100 ***[1 mark]***

ii) From CAT, 43 = C and from DOG, 44 = D and 4F = O.
E is 1 more than D (= 44) so 45 = E. ***[1 mark]***
The password is CODE. ***[1 mark]***

5 a) denary(A) = 10 and denary(C) = 12 ***[1 mark]***
So, denary(A) + denary(C) = 10 + 12 = 22 ***[1 mark]***

b) Taking a hexadecimal as an input. ***[1 mark]***
Splitting hexadecimal into characters. ***[1 mark]***
Multiplying first character in denary by 16. ***[1 mark]***
Adding the second character in denary to the first. ***[1 mark]***
E.g.

```
hexadecimal = input("Enter a two digit hexadecimal")
char1 = hexadecimal[0]
char2 = hexadecimal[1]
ans = 16 * denary(char1) + denary(char2)
print(ans)
```

Page 114 (Warm-Up Questions)

1 ASCII uses fewer bits to represent each character but is a much smaller character set.

2 16

3 a) The file size would increase.

b) The file size would decrease.

4 200 kbit/s

5 a) The file size and quality would both increase.

b) The file size and quality would both decrease.

6 E.g. Lossy compression will make file sizes a lot smaller. Lossless compression will maintain the quality of the original file.

Page 115 (Exam Questions)

3 a) A character set is a collection of characters a computer recognises from their binary representation. ***[1 mark]***

b) E.g. Binary code sent to computer. ***[1 mark]***
Character set used to translate binary code. ***[1 mark]***

4 a) Extract 2 would have a better sound quality ***[1 mark]*** because it has a larger bit rate and sampling frequency, both of which are indications of better overall sound quality. ***[1 mark]***

b) The file size would be larger. ***[1 mark]***

5 a) 60 × 60 × 10 × 10 ***[1 mark]*** = 360 000 ***[1 mark]***

b) A lower DPI would reduce the quality of the image ***[1 mark]*** because the number of pixels in a given area decreases. ***[1 mark]***

c) Metadata is 'data about data' / information about a file. ***[1 mark]*** Metadata includes information about height, width, resolution etc. so the image is displayed properly. ***[1 mark]***

Page 116 (Revision Questions)

2

A	B	C	A OR B	(A OR B) AND C
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

5 a) 2 terabytes

b) 2 000 000 megabytes

6 a) i) 10001 ii) 10010100 iii) 11110000

b) i) 11 ii) 94 iii) F0

7 a) i) 56 ii) 159 iii) 43

b) i) 38 ii) 9F iii) 2B

8 a) i) 74 ii) 117 iii) 3033

b) i) 1001010 ii) 1110101 iii) 101111011001

9 10001111

Practice Paper 1 — Computer Systems

1 a) i) Utility software is software that helps to configure, optimise or maintain a computer. ***[1 mark]***

ii) Any **two** examples of utility software, e.g.
- Disk defragmentation software ***[1 mark]***
- System diagnostic tools ***[1 mark]***
- Anti-virus / anti-spyware ***[1 mark]***
- Backup software ***[1 mark]***
- Compression software ***[1 mark]***
- File management software ***[1 mark]***

[2 marks available in total]

b) i) E.g. When the hard disk is fragmented, it will take longer to read/write data on the hard disk. ***[1 mark]*** This in turn may slow down the computer. ***[1 mark]***

ii) E.g.
- Defragmentation software reduces fragmentation by moving files on the hard disk. ***[1 mark]***
- The empty spaces/gaps are collected together. ***[1 mark]***
- Different bits of the same file are moved to be stored together. ***[1 mark]***
- This means the read/write heads won't need to move as far across the disk, so the read/write speed should improve. ***[1 mark]***

[3 marks available in total]

c) E.g. Disk clean-up will remove a number of files, so it will immediately leave gaps in the data stored on the hard disk, leading to fragmentation. ***[1 mark]***

d) Any **three** advantages, e.g.
- SSDs are faster than HDDs. ***[1 mark]***
- SSDs do not need to be defragmented. ***[1 mark]***
- SSDs do not make any noise. ***[1 mark]***
- SSDs are more shock-proof than HDDs. ***[1 mark]***

[3 marks available in total]

2 a) The use of online servers provided by a hosting company to store files and software. ***[1 mark]***

b) E.g.
- Laptop/web browser sends a request to the cloud server, to send the image. ***[1 mark]***
- The cloud server processes the request. ***[1 mark]***
- The cloud server replies with the image. ***[1 mark]***

[2 marks available in total]

c) i) E.g.
- The cloud server splits the image into packets. ***[1 mark]***
- Each packet is given control information, including the IP addresses of the server and the laptop. ***[1 mark]***
- The server calculates a checksum number for each packet. ***[1 mark]***
- Each packet is given a packet number to show the order of the data. ***[1 mark]***
- Each router reads the control information of the packet and decides which way to send the data. ***[1 mark]***
- IP protocol governs packet switching. ***[1 mark]***
- The way the data is sent changes according to network traffic so packets take different routes to their destination. ***[1 mark]***
- Packets arrive at their destination in the wrong order. ***[1 mark]***
- The receiving device puts the packets in the right order using the packet numbers on each packet. ***[1 mark]***
- The receiving device recalculates the checksum. ***[1 mark]***
- If all packets are successfully delivered and the checksums match, a receipt confirmation is sent back to the sending device. ***[1 mark]***

[6 marks available in total]

ii) E.g.
- There are many possible routes the data can take ***[1 mark]*** so the data can reach the destination even if there is heavy network traffic. ***[1 mark]***
- Splitting data into small packets allows different parts of the data to be routed separately ***[1 mark]*** which is more flexible than sending the whole file in one go. ***[1 mark]***

[2 marks available in total]

iii) Internet Protocol (IP) ***[1 mark]***

d) i) A set of rules and procedures the organisation will follow to ensure their network is protected against attacks and unauthorised access. ***[1 mark]***

ii) E.g.
- Network forensics are investigations undertaken to find the cause of attacks on a network. ***[1 mark]***
- To conduct network forensics, an organisation needs to have a system of capturing data packets as they enter their network. ***[1 mark]***
- After the network is attacked, data packets can be analysed to discover how the network was attacked. ***[1 mark]***
- The information gained from network forensics can be used to decide how to prevent future attacks. ***[1 mark]***

[3 marks available in total]

3 a) i) Ethernet is a network protocol used on wired networks. ***[1 mark]***
WPA2 is a security protocol used on wireless LANs. ***[1 mark]***

ii) Any **one** difference, e.g.
- CAT5e twisted pair cables use four twisted copper wires ***[1 mark]*** whereas coaxial cable uses one single copper wire. ***[1 mark]***
- CAT5e cables prevent interference by twisting the wires together ***[1 mark]*** whereas coaxial cable uses a braided metallic shield to prevent interference. ***[1 mark]***

[2 marks available in total]

iii) E.g.
The Leeds studio's wired setup:
- Wired connections have a more reliable performance as there is no loss of signal no matter where the devices are in the building. ***[1 mark]***
- Wired connections are more restrictive as it is harder to add new devices / access the network while moving through the building. ***[1 mark]***

The York studio's wireless setup:
- Wireless connections are easier for the employees to connect to (e.g. no need for cables to add a laptop or mobile device to the network). ***[1 mark]***
- Wireless connections can suffer from signal problems caused by building interference or interference from other wireless signals nearby. ***[1 mark]***

To get all four marks, you'll need one advantage and one disadvantage for wired connections, as well as one advantage and one disadvantage for wireless connections.

b) i) Any **one** advantage, e.g.
- Fibre optic cables tend to have greater bandwidth / can carry more data than copper cables. ***[1 mark]***
- Fibre optic cables can carry data over longer distances / don't suffer signal degradation or interference. ***[1 mark]***
- Fibre optic cables are easier to maintain than copper cables so cost less in the long term. ***[1 mark]***

[1 mark available in total]

ii) E.g.
- Laying its own cables between Leeds and York could be too expensive for the company. ***[1 mark]***
- Leased lines are likely to be more reliable and faster than other WAN connections. ***[1 mark]***

[1 mark available in total]

4 Points you might include:

Stakeholders

- Consumers will be able to access a greater variety of books, films and music. They will be able to purchase them at their own convenience without having to go to a shop. The books, films and music should also be cheaper as the costs are much lower to produce digital copies than physical copies.
- Traditional bookshops, CD shops and film retailers will have to adapt to distributing their stock digitally or face a decline in profits.
- Content creators will be able to distribute their media more cheaply, as they no longer have to pay for a record label, production team or publisher.
- New businesses that distribute books, music and films digitally could flourish as they gain some of the traditional stores' market share. New business models have been founded to distribute digital media, e.g. subscription-based services or streaming services where money is made through advertising.

Technology

- Use of digital media has encouraged the development and improvement in technology. E.g. e-readers introduced a new type of display technology that looks and feels more like a printed book. In the future, displays which are even more paper-like may be used.
- The use of digital media has placed extra demands on communication networks. This creates a cycle of improvement, e.g. watching high-quality films increases demand for better broadband speeds, which in turn encourages people to watch higher quality films, increasing demand for good broadband.
- The use of digital media is changing TVs from devices that receive broadcasted transmissions to Internet-enabled devices used to stream media from the Internet.

Environmental issues

- Less use of physical media means there could be less physical waste from packaging and discarded books, CDs and DVDs.
- No need for packaging or physical media, so fewer natural resources required in the distribution of the media.
- Greater use of digital media means increased use of electricity, as well as increased e-waste if hardware is discarded.

Legal issues

- Use of digital media could increase the amount of illegal file sharing.
- It is often unclear who owns the copyright to the original piece of work, and how profit from the book / music / film should be shared. E.g. how much money should a musician receive from their music being listened to on a free streaming service.
- It is often unclear who owns the digital file. E.g. should someone be able to pass their movie files onto their children, or does the ownership over them go back to the original distributor.

How to mark your answer:

- Two or three brief points with very little explanation. ***[1-2 marks]***
- Three to five detailed points covering at least two of: stakeholders, technology, environmental issues and legal issues. ***[3-5 marks]***
- Six or more detailed points that form a well-written, balanced discussion, covering all of: stakeholders, technology, environmental issues and legal issues. ***[6-8 marks]***

5 a) Storage ***[1 mark]***, GPU ***[1 mark]***

b) E.g. Hardeep should upgrade the RAM in his computer, ***[1 mark]*** as he only just has enough RAM to cover the minimum requirements, so the OS may not run very smoothly. ***[1 mark]***

It's fine to answer 'no', as long as you've justified your answer.

c) RAM stores applications and data that is currently in use. ***[1 mark]*** As operating systems are running all the time, a large amount of the OS is kept in RAM. ***[1 mark]***

d) Any **two** features, e.g.

- Large buttons and icons ***[1 mark]*** that can be pressed to open applications and windows. ***[1 mark]***
- Screens and menus ***[1 mark]*** that can be navigated / controlled by swiping or dragging with a finger. ***[1 mark]***
- Support for finger gestures ***[1 mark]*** such as pinching to zoom out / tap and hold to open additional options / four finger swipes to swap between apps etc. ***[1 mark]***
- Virtual on-screen keyboard ***[1 mark]*** to allow the user to type without attaching an external keyboard. ***[1 mark]***

[4 marks available in total]

6 a)

	Allowed	Not Allowed
Giving each teacher a USB flash drive containing the personal information of all the pupils who have attended the school.		✓
Transferring pupils' personal information to their new school when they leave.	✓	
Refusing to tell a parent what information is being stored about their son/daughter.		✓
Backing up pupils' personal information on a drive that is stored in a locked safe.	✓	
Using pupils' personal information to get in contact with their parents.	✓	
Putting pupils' personal information on the school website to make it easier for teachers and parents to access.		✓

[6 marks available — 1 for each correct row]

b) Any **two** forms of network attack and a suitable method of protection, e.g.

- Passive attack ***[1 mark]*** — using tools such as network-monitoring devices and packets sniffers to intercept sensitive information. ***[1 mark]*** Can be protected against by encrypting data on the network. ***[1 mark]***
- Active attack ***[1 mark]*** — using malware or other means to break into a network and steal data. ***[1 mark]*** Can be protected against by using a firewall. ***[1 mark]***
- Insider attack ***[1 mark]*** — exploiting a person's position within the organisation to gain access to data on the network. ***[1 mark]*** Can be protected against by using user access levels to restrict access to people that are trusted. ***[1 mark]***
- Brute force attack ***[1 mark]*** — using automated software and trial-and-error to find a password that will allow access to the network. ***[1 mark]*** Can be protected against by locking accounts after a certain number of failed attempts. ***[1 mark]***

[6 marks available in total]

7 a) If the central switch fails, users will be unable to access files on other servers ***[1 mark]*** because there will be no connections between any of the servers. ***[1 mark]***

b) E.g.

- Connecting servers in a mesh network means that if one connection fails, there are other routes to connect the servers. ***[1 mark]*** This means the network can still operate fully and all the computers can access data on all servers. ***[1 mark]***
- Adding an extra server to a mesh network would be more complicated ***[1 mark]*** because it would need to be connected to all of the other servers rather than just the switch. ***[1 mark]***
- Some connections would be redundant most of the time ***[1 mark]*** as the servers would communicate using the quickest or most direct route. ***[1 mark]***

[4 marks available in total — at least one advantage and one disadvantage required for full marks]

c) Increasing incTotal and fullTotal by the number of files edited on that day. ***[1 mark]***
Conditional statement based on the values of fullTotal and incTotal. ***[1 mark]***
Taking full backup if fullTotal reaches 1000 (regardless of the value of incTotal). ***[1 mark]***
Resetting fullTotal and incTotal to 0 after full backup. ***[1 mark]***
Taking incremental backup if incTotal reaches 200 (unless fullTotal ≥ 1000). ***[1 mark]***
Resetting only incTotal to 0 after incremental backup. ***[1 mark]***
E.g.

```
dailyTotal = number of files edited today
incTotal = incTotal + dailyTotal
fullTotal = fullTotal + dailyTotal
if fullTotal >= 1000 then
    take full backup
    fullTotal = 0, incTotal = 0
elseif incTotal >= 200 then
    take incremental backup
    incTotal = 0
endif
```

Practice Paper 2 — Computational Thinking, Algorithms and Programming

1 a) i) String ***[1 mark]***

ii) Real/Float ***[1 mark]***

b) Using the correct data type will make your programs more memory efficient. ***[1 mark]*** It will also make your programs more robust and predictable. ***[1 mark]***

c) E.g. All the different data values might be converted to strings using the str() function so string concatenation can be used to join them together on the receipt. ***[1 mark]*** For example the cost of fuel might be converted from a real / float to a string so it can be printed in a sentence, e.g. "The cost of fuel was £20.76". ***[1 mark]***

d) Any **two** differences, e.g.
- A compiler would produce an executable file ***[1 mark]*** while an interpreter would not. ***[1 mark]***
- A compiler would list any errors at the end of the translation ***[1 mark]*** while an interpreter would return the first error it found then stop. ***[1 mark]***
- A compiler would translate the code all at once ***[1 mark]*** while an interpreter would translate and run the code line by line. ***[1 mark]***

[4 marks available in total]

2 a) 10101000 = 128 + 32 + 8 = 168
[2 marks available — 1 for correct working and 1 for correct final answer]

b) Split 10110101 into nibbles:
1011 = B, 0101 = 5, so 10110101 = B5
[2 marks available — 1 for correct working and 1 for correct final answer]

c)
```
  10101000
+ 10110101
-----------
 101011101
 1
```
[1 mark]

d) An overflow error is where the result of a calculation requires more bits than are available and some of the data is lost. ***[1 mark]***

3 a) i) A parameter is a special variable that passes data into a sub program. ***[1 mark]***

ii) Parameters have a local scope to the sub program they're defined in. ***[1 mark]***

iii) Arguments are the actual values that parameters take when the sub program is called. ***[1 mark]***

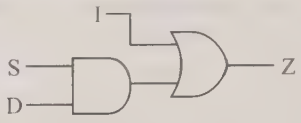
b) i) Declares a new variable, difficulty, ***[1 mark]*** which has global scope so it can be used anywhere in the program. ***[1 mark]***

ii) It takes weight as a parameter ***[1 mark]*** and sets the variable difficulty equal to the integer (quotient) part when weight is divided by 6. ***[1 mark]***

c) A sub program that takes heart rate as a parameter. ***[1 mark]***
Using a selection statement. ***[1 mark]***
Having correct conditions on the selection statement. ***[1 mark]***
Setting the difficulty depending on the conditions. ***[1 mark]***
E.g.

```
procedure adjustLevel(heartrate)
  if heartrate > 160 then
    difficulty = 0
    print("Slow Down!")
  elseif heartrate < 90 then
    difficulty = difficulty + 1
  elseif heartrate > 140 then
    difficulty = difficulty - 1
  endif
endprocedure
```

4 a)



[3 marks available — 1 for inputs S and D going into an AND gate, 1 for the output of AND gate and input I going into an OR gate and 1 for Z being the OR gate output]

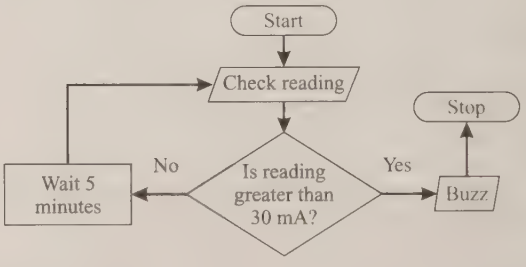
b)

S	D	I	Z
[illegible]	[illegible]	0	[illegible]
1	1	[illegible]	1
[illegible]	0	0	[illegible]

[3 marks available — 1 for each correct row]

5 a) Check each item in order.
Check the first item: 10 mA ≠ 12 mA
Check the second item: 15 mA ≠ 12 mA
Check the third item: 12 mA = 12 mA.
Stop searching as the item has been found.
[2 marks available — 1 for starting with 10 mA and 1 for checking items in order until you find 12 mA]

b) Using Start / Begin. ***[1 mark]***
Input to check reading. ***[1 mark]***
Decision box with appropriate question. ***[1 mark]***
'Buzz' as an output. ***[1 mark]***
Wait 5 minutes as a process. ***[1 mark]***
Using Stop / End. ***[1 mark]***
E.g.



6 a) 16^6 ***[1 mark]***

b) E.g. It would be easier to remember the hex code for a particular colour ***[1 mark]*** because the hex code would be shorter than the binary or denary equivalents. ***[1 mark]***

c) JPEG is an example of lossy compression ***[1 mark]*** — it produces a much smaller file size than the other formats. ***[1 mark]***

d) E.g.
- No data is lost when compressed ***[1 mark]*** so graphics can be reverted back to the original — this is essential to ensure graphics are always high quality. ***[1 mark]***
- File sizes are only slightly reduced ***[1 mark]*** so there would be an impact on storage requirements to the company. ***[1 mark]***
- Not all software is compatible with lossless file types ***[1 mark]*** so clients may not be able to open graphics. ***[1 mark]***
- File sizes are large ***[1 mark]*** so the images will take longer to upload or attach to emails for clients. ***[1 mark]***

[4 marks available in total]

e) Any **two** pieces of metadata, e.g.
- File format ***[1 mark]***
- Height/width in pixels ***[1 mark]***
- Colour depth ***[1 mark]***
- Resolution ***[1 mark]***
- Date created/date modified ***[1 mark]***
- Creator's name ***[1 mark]***

[2 marks available in total]

7 a) Erroneous test data contains values that the program should not accept. ***[1 mark]*** Using them in testing will make sure that the user can enter any data without breaking the program. ***[1 mark]***

b) i) Type of data: Normal ***[1 mark]***
Intended Outcome: 20 ***[1 mark]***
Actual Outcome: 20 ***[1 mark]***

ii) Type of data: Erroneous ***[1 mark]***
Intended Outcome: User receives an error message, e.g. "Invalid dice score, please re-enter." ***[1 mark]***
Actual Outcome: 19 ***[1 mark]***
The game uses a six-sided dice, so if the user inputs a 9 they should receive an error message. However, Tony's program does not check for this.

c) Any **two** ways, e.g.
- He could make sure only integers are accepted. ***[1 mark]***
- He could make sure only the numbers from 1 to 6 are accepted. ***[1 mark]***
- He could ensure that the input is only one character long. ***[1 mark]***
- He could make sure that only Yes/No, yes/no, Y/N, y/n, etc. are accepted for the "Roll again" prompt. ***[1 mark]***

[2 marks available in total]

8 a) E.g. Storing the data in records allows the different fields to have different data types ***[1 mark]*** while if they used an array, they would all have to be the same data type. ***[1 mark]***

b) i) SELECT Title FROM comics
WHERE Genre = "Science Fiction"
[2 marks available — 1 for selecting the correct fields from the comics table and 1 for a correct 'where' statement]

ii) SELECT Title, Length FROM comics
WHERE Length < 50 AND Rating = 3
[2 marks available — 1 for selecting the correct fields from the comics table and 1 for a correct 'where' statement]

iii) SELECT * FROM comics WHERE Title LIKE "H%"
[2 marks available — 1 for selecting all fields from the comics table and 1 for a correct 'where' statement]

9 a) array primeKoalas[4, 2]
primeKoalas[0, 0] = "John"
primeKoalas[1, 0] = "Paul"
primeKoalas[2, 0] = "Cheryl"
primeKoalas[3, 0] = "Ida"
primeKoalas[0, 1] = "guitar"
primeKoalas[1, 1] = "bass"
primeKoalas[2, 1] = "vocals"
primeKoalas[3, 1] = "drums"
[2 marks available — award 2 for all 4 correct lines of code, otherwise award 1 for at least two correct lines of code]

b) Opening and closing the file properly. ***[1 mark]***
Using a loop. ***[1 mark]***
Writing the name and instrument of each band member to the file. ***[1 mark]***
E.g.
```
band = openWrite("musicians.txt")
for i = 0 to 3
  band.writeLine(primeKoalas[i, 0] + " " + primeKoalas[i, 1])
next i
band.close()
```

c) A procedure taking a file name as a parameter. ***[1 mark]***
Opening the file in read mode. ***[1 mark]***
A condition-controlled loop to stop at the end of the file. ***[1 mark]***
Using the toSpeech() procedure to read out each line. ***[1 mark]***
Closing the file. ***[1 mark]***
E.g.
```
procedure readAll(fileName)
  file = openRead(fileName)
  while NOT file.endOfFile()
    toSpeech(file.readLine())
  endwhile
  file.close()
endprocedure
```
It's important that you always remember to close a file once you're finished reading from it or writing to it.

Glossary and Index

A

abstraction Picking out the important bits of information. **51**

accumulator The part of the ALU that stores the intermediate results when doing a calculation. **3**

active attack (networks) A network attack where the hacker attempts to change data or introduce malware. **31**

algorithmic thinking Coming up with an algorithm to solve a problem. **51**

algorithm A step-by-step set of rules or instructions. **51-59**

alphanumeric The collective name for letters, digits and symbols. **67, 110**

analogue signal A continuous signal which can't be processed by a computer. **112**

AND One of the Boolean operators.
logic gate 98
operator 73

anti-malware software Software designed to stop malware from damaging a computer or network. **33**

application A piece of software written to help users do various tasks, often through a graphical user interface with menus and toolbars. **1, 6, 12, 13**

architecture Describes how the CPU works and interacts with the other parts of the computer system. **2, 3**

argument A value that a parameter of a sub program actually takes. **81**

arithmetic logic unit (ALU) The part of the CPU that carries out arithmetic and Boolean operations. **2, 3**

arithmetic operator An operator that programs use to carry out simple mathematical operations. **64**

array A data structure where all the data is stored and defined under one variable name. **76, 77**
one-dimensional 76
two-dimensional 77

ASCII A 7-bit character set consisting of 128 characters. **110**

assembler A program that turns assembly language into machine code. **92**

assembly language A low-level language. **92**

assignment operator (=) Assigns the value on the right hand side to the name on the left hand side. **65**

authentication A process for checking the identity of the user. **88**

auto-documentation A programming tool commonly used to make a summary of a program. **93**

B

backdoor A vulnerability in a computer or network's security that a hacker could exploit. **31**

backup utilities 15

bandwidth The amount of data that can be transferred on a network in a given time. **20, 113**

binary A counting system using base-2 consisting of 0s and 1s. **103-105**
addition 104
converting to/from denary 103
converting to/from hexadecimal 107
overflow 104
shifts 105

binary search algorithm 56

binary shift Moving the bits in a binary number left or right and filling the gaps with 0s. **105**

BIOS (Basic Input Output System) Software stored in ROM responsible for booting up a computer system. **6**

bitmap image A graphic that is made up of pixels. **111**

bit A binary digit, either 0 or 1. **100**

bit rate The number of bits used per second of sampled audio. **112**

Boolean A logical system using the operators OR, AND and NOT. The Boolean data can take one of two values, either true or false.
data type 63
logic gates 98, 99
operators 73

breakpoint A programming tool used to halt a program at a specific place. **93**

brute force attack A network attack which uses software to crack security passwords through trial and error. **31**

bubble sort algorithm 57

bus topology A network topology in which all devices are connected to a single backbone cable. **24**

byte 8 bits. **100**

C

cache (CPU) Quick access memory inside the CPU. **2, 7**

casting A way of changing from one data type to another. **64**

cat 5e / cat 6 cable Standards for copper Ethernet cables used on LANs. **21**

censorship The control (usually by a government or organisation) of what information other people can access. **39**

channel (Wi-Fi®) A small range of Wi-Fi® frequencies. **22**

character A single alphanumeric symbol. **110**
(as a data type) 63

character set A group of characters that a computer recognises from their binary representation. **110**

check digit A digit added to a string of digits which is used to check if all the digits have been entered and read correctly. **100**

checksum A number used to check if a packet of data sent between networks has been received correctly. **28**

client A device which requests data from a server. **23**

client-server network A type of network managed by a server, which takes requests from client devices. **23**

clock speed The number of instructions a processor can carry out each second. **7**

cloud computing The use of the Internet to store files and provide software. Also known as the cloud. **30, 39**

code editor Part of an IDE where you write and edit your source code. **93**

colour depth The number of bits used for each pixel in an image file. **111**

command-line interface A text-based user interface where the user types in commands. **13**

comment A note added to source code to say what part of a program does. **89**

comparison operator Compares two values and outputs either true or false. **65**

compiled code An executable file created by a compiler. **92**

compiler A programming tool to translate source code into machine code. **92**

compression The process of making the size of a file smaller. **15, 113**

computational thinking Tackling a problem through decomposition, abstraction and algorithmic thinking. **51**

Computer Misuse Act 46

concatenation Joining strings together. **67**

condition-controlled loop An iteration statement that repeats a set of instructions until a condition is met. **72**

constant A named value which cannot be altered as the program is running. **66**

control unit (CU) The part of the CPU that controls the flow of data. **2**

copyright A legal right that prevents others from copying or modifying your work without permission. **47**

Copyright, Designs and Patents Act 47

core (CPU) A processing unit found inside the CPU. **7**

Glossary and Index

count-controlled loop An iteration statement that repeats a set of instructions a given number of times. **71**

CPU (or processor) The part of the computer system that processes the data. It contains the control unit, ALU and cache. **1-3**
performance 7

Creative Commons licence Permits sharing of a creative work as long as certain requests from the original creator are upheld. **47**

cultural issue An issue which affects a particular religious, ethnic, national or other group. **38-42**

cyberbullying Using social media to deliberately harm someone else. **40**

D

database A collection of data records (made up of fields) often represented as tables. **80**

Data Protection Act 46

data type Tells you what kind of data it is, e.g. integer, real, string, etc. **63**

debugging Identifying and fixing errors in a program. **93**

decomposition Breaking a problem down into smaller problems. **51**

dedicated system A computer system designed to carry out a specific task. **1**

defensive design A method of designing a program so that it functions properly and doesn't crash. **87-89**

defragmentation Reorganising data on a hard drive to put broken up files back together and collect up the free space. **15**

denary A number system using base-10. Also known as decimal. **103**
converting to/from binary 103
converting to/from hexadecimal 106

denial-of-service attack A network attack which stops users from accessing a part of a network or website. **31**

Designs and Patents Act 47

device driver A piece of software that allows applications to communicate with a piece of hardware. **12**

digital divide The inequality created by the fact that some people have greater access to technology than others. **42**

digital signal The representation of an analogue signal using binary data. **112**

disk management Organisation and maintenance of the hard disk. **14, 15**

domain name server (DNS) A server which stores website domain names and their IP addresses. **30**

dongle A small piece of hardware which allow devices to connect to a network wirelessly. **22**

DO UNTIL loop Type of iteration statement. **72**

DO WHILE loop Type of iteration statement. **72**

dynamic IP address An IP address which is automatically assigned to a device when it connects to a network. **27**

E

ELSEIF 70

embedded system A computer built into another device, e.g. a Smart TV. **1**

encryption Coding ('encrypting') data so that it can only be decoded ('decrypted') with the correct key. **15, 33**

environmental issue An issue relating to how we impact the natural world. **38, 45**

erroneous data Test data that a program isn't designed to accept. **91**

error diagnostics Information about an error once it's been detected. **93**

errors (programming) 90

Ethernet Network protocol used on LANs. **27**

ethical issue Something which raises questions of right and wrong. **38-42**

E-waste Discarded computer material. **45**

extended ASCII An 8-bit character set consisting of 256 characters. **110**

extreme data Test data on the boundary of what the program will accept. **91**

F

fetch-decode-execute cycle The process that the CPU uses to retrieve and execute instructions. **3**

fibre optic cable A high performance cable that uses light to carry data. **20, 21**

field An element of a record used to store one piece of data. A column of a database table. **79, 80**

file handling (programming) Reading from and writing to external files. **78**

file management The organisation, movement and deletion of files. **14**

file sharing Copying files between devices on a network. **23, 47**

final testing When the testing stage of the software development cycle is only done once to check the software meets all the initial requirements. **91**

firmware Permanent software stored on ROM, used to control hardware or embedded systems. **6**

flash memory Solid state non-volatile data storage. **6, 8, 9**

flow diagram A graphical way of showing an algorithm. **53**

FOR loop A type of count-controlled iteration statement. **71**

frame The unit used to send data via Ethernet over a LAN. **21**

Freedom of Information Act 46

FTP (File Transfer Protocol) A protocol used to access, edit and move files on another device, like a server. **29**

functionality testing A type of testing that assesses how well a program meets the requirements. **90**

function A sub program that takes parameters and returns a value. **81, 82**

G

gigabyte 1000 megabytes. **100**

global divide The digital divide between different countries. **42**

global variable (programming) A variable available throughout the whole program. **82**

GPU (Graphics Processing Unit) A circuit for handling the processing of graphics and images. **7**

graphical user interface (GUI) Allows the user to interact with the computer in a visual and intuitive way. **13**

graphics card A piece of hardware containing a GPU. **7**

GUI builder An IDE tool for giving a program a graphical user interface. **93**

H

hacker A person who tries to illegally access or attack a computer network or device. **31, 46**

hard disk drive (HDD) Traditional internal storage for PCs and laptops that stores data magnetically. **1, 8, 9**

hardware The physical parts of a computer system. **1, 12, 21**

heat sink Pulls heat away from the CPU to help maintain its temperature. **1**

hexadecimal A counting system using base-16 consisting of the digits 0-9 and the letters A-F. **106, 107**
converting to/from binary 107
converting to/from denary 106

high-level language A programming language like C++ and Java™ that is easy for humans to understand. **92**

Glossary and Index

hosting (Internet) When a business uses its servers to store the files of another organisation. **30**

hotspot A location where people can access a wireless access point. **22**

HTTP (Hyper Text Transfer Protocol) Used by web browsers to access websites and communicate with web servers. **29**

I

IDE (Integrated Development Environment) A piece of software to help a programmer develop programs. **93**

IF statement Type of selection statement. **70**

images 111

IMAP (Internet Message Access Protocol) A protocol used to retrieve emails from a server. **29**

incremental backup A record of all the changes since the last incremental backup. **15**

indentation Spaces put at the beginning of lines of code to help show a program's structure. **89**

input sanitisation Removing unwanted characters from an input. **87**

input validation Checking that an input meets certain criteria. **32, 87**

insertion sort algorithm 59

insider attacks A network attack where someone within an organisation exploits their network access to steal information. **31, 33**

integer (data type) A numerical data type for whole numbers. **63**

intellectual property An original piece of work (or an idea) that someone has created and belongs to them. **47**

Internet The biggest WAN in the world, based around the TCP/IP protocol. **30, 39-42**

Internet Protocol (IP) The protocol responsible for packet switching. **29**

interpreter A translator that turns the source code into machine code and runs it one instruction at a time. **92**

IP address A unique identifier given to a device when it accesses an IP network. **27**

iteration statement A statement which makes the program repeat a set of instructions. **71, 72**

iterative testing Repeated testing done during the development of a program. **91**

K

kilobyte 1000 bytes. **100**

L

LAN (Local Area Network) A network which only covers a single site. **20**

lawful interception Checking data on a network for cyber security purposes. **31**

layers (network) Groups of protocols that have similar functions. **29**

legal issue An issue relating to what's right and wrong in the eyes of the law. **38, 46, 47**

linear search algorithm 56

linker A programming tool which can combine different compiled codes. **92**

Linux An open source operating system. **16**

local variable A variable that is only defined and usable within certain parts of a program. **82**

logic circuit An electronic circuit for performing logic operations on binary data. It may have more than one logic gate and more than two inputs. **99**

logic error When a program does something that was not intended. **90**

logic gate An electronic circuit component that performs a Boolean operation (e.g. AND, OR or NOT). **98**

loop (programming) A set of instructions that the program repeats until a condition is met or count is reached. **71, 72**

lossless compression Temporarily removing data from a file to decrease the file size. **113**

lossy compression Permanently removing data from the file to decrease the file size. **113**

low-level language A programming language that is close to what a CPU would actually do and is written for specific hardware (i.e. CPU type). E.g. machine code and assembly languages. **92**

M

MAC address A unique identifier assigned to a device that cannot be changed. **27**

machine code The lowest-level programming language consisting of 0s and 1s. CPUs can directly process it as a string of CPU instructions. **92**

magnetic storage Hard disk drives and magnetic tapes that hold data as magnetised patterns. **8, 9**

mainframe (or supercomputer) An extremely powerful (and expensive and reliable) computer for specialist applications. **1, 14**

maintainability A characteristic of defensive design that helps programmers modify and repair programs. **89**

malware Malicious software created to damage or gain illegal access to computer systems. **31, 33**

megabyte 1000 kilobytes. **100**

memory Hardware used to store data that a CPU needs access to. **3, 6**

memory address register (MAR) A CPU register that holds memory addresses (locations) for data and instructions that the CPU needs. **3**

memory data register (MDR) A CPU register that holds data and instructions. **3**

merge sort algorithm 58

mesh topology A network topology where every device is directly or indirectly connected to every other without a central switch or server. **24**

metadata Extra data stored in a file which gives information about the file's properties. **111**

motherboard The main circuit board in a computer that other hardware connects to. **1, 6, 21**

multi-tasking (OS) When an operating system runs multiple programs and applications at the same time. **13**

multi-user (OS) When an operating system allows multiple users at the same time. **14**

N

nested IF statement A selection statement made up of multiple IF statements inside each other. **70**

network forensics Investigations that organisations undertake to find the cause of attacks on their network. **33**

network interface controller (NIC) An internal piece of hardware that allows a device to connect to a network. **21**

network policy A set of rules and procedures an organisation will follow to ensure their network is protected against attacks. **33**

network security Protection against network attacks. **31-33**

nibble 4 bits. **100**

non-volatile memory Memory that retains its contents when it has no power. **6**

Glossary and Index

normal data A type of test data that simulates the inputs that users are likely to enter. **91**

NOT One of the Boolean operators.
logic gate 98
operator 73

O

open source (software) Software that can be modified and shared by anyone. **16**

operating system (OS) A piece of software responsible for running the computer, managing hardware, applications, users and resources. **12-14**

operator A special symbol like +, *, =, AND, ==, that carries out a particular function. **64, 65**

optical disc CD, DVD or Blu-Ray™ disc that is read / written to with lasers. **9**

optical drive Device used to read and write to optical discs. **1**

OR One of the Boolean operators.
logic gate 98
operator 73

overclocking Running a CPU at a higher clock speed than was intended. **7**

overflow error An error that occurs when the computer attempts to process a number that has too many bits for it to handle. **104**

P

packets (networks) Small, equal-sized units of data used to transfer files over networks. **28, 29**

packet switching The process of directing data packets on a network using routers and the IP protocol. **28**

parameter A variable that a sub program requires in order to run — it's only defined within the sub program. **81**

parity bit A bit placed at the end of binary data to show if it's been received correctly. **100**

passive attack (networks) Where a hacker monitors data travelling on a network. **31**

password A string of characters that allows access to certain parts of a computer or program. **14, 33, 88**

patent A licence that protects new inventions, ideas and concepts. **47**

payload (network packets) The part of a packet with the actual data. **28**

peer-to-peer (P2P) network A network in which all devices are equal and connect directly to each other. **23**

pentesting (penetration testing) The process of simulating attacks on a network to identify weaknesses. **33**

peripherals External hardware connected to a computer. **1, 12**

petabyte 1000 terabytes. **100**

phishing When criminals send emails or texts to someone claiming to be a well-known business. **32**

pixels Small dots that make up a bitmap image. **111**

platform (OS) A computer system that other applications can run on. **13**

POP3 A protocol used to retrieve emails from a server. **29**

power supply A piece of hardware that gives the other pieces of hardware the energy they need to run. **1**

primary storage Memory that can be accessed directly by the CPU. **6, 8**

privacy 39

procedure A sub program that carries out a list of instructions. **81, 82**

processing The execution of program instructions by the CPU. **1-3**

program counter (PC) Holds the memory address of the next CPU instruction. **3**

program flow The order in which statements are executed in a program (controlled with selection and iteration statements). **70-72**

program A set of instructions that can be executed on a computer.
programming 63-82

proprietary software Software where modifying and sharing is not permitted. **16**

protocols (networks) A set of rules for how devices communicate over a network. **27-29**

pseudocode A set of instructions in the style of a programming language but using plain English. **52**

public domain Describes content which has no copyright attached to it. **47**

Q

Query A request to retrieve data that meets certain conditions from a database. **80**

R

RAM The main memory of a computer. **1, 6, 7**

ransomware A type of malware that uses encryption to lock a user out of their files. **31**

real (data type) A numerical data type for decimal numbers. **63**

record A data structure used to store multiple items of data about one 'thing' together. A row in a database table. **79, 80**

register A temporary data store inside a CPU. **2, 3**

repetitive strain injury (RSI) A health problem caused by doing repeated movements over a long period of time. **41**

resolution The density of pixels in an image, often measured in dpi. **111**

ring topology A network topology where the devices are connected in a ring, with data moving in one direction. **24**

ROM (Read only memory) Memory that can be read but not written to. **6**

router A piece of hardware responsible for transmitting data between networks. **21, 28**

run-time environment Allows code to be run and tested from within an IDE. **93**

S

sample size The number of bits available for each audio sample. **112**

sampling The process of converting analogue signals to digital data. **112**

sampling frequency The number of audio samples that are taken per second. **112**

sampling intervals The time between each sample. **112**

sanitisation (programming) Removing unwanted characters from an input. **87**

scareware A type of malware that creates false messages to trick the user into following malicious links. **31**

script A simple program, often run on command-line interfaces to automate tasks. **13**

search algorithm A set of instructions that you can follow to find an item in a list. **56**

secondary storage External data storage used to store data so that the computer can be switched off. **6, 8, 9**

selection statement A statement which causes the program to make a choice and flow in a given direction — e.g. IF and SWITCH-CASE statements. **70, 71**

Glossary and Index

server A device which provides services for other devices (clients), e.g. file storage / web pages / printer access. **23, 45**

sharing economy Where people make money from things they already own. **42**

single-user (OS) When an operating system only allows one user to use it at any one time. **14**

SMTP (Simple Mail Transfer Protocol) Used to send emails and transfer emails between servers. **29**

social engineering A way of gaining illegal access to data or networks by influencing people. **32**

social media Web applications which allow people to communicate and share content with others online. **39-41**

software Programs or applications that can be run on a computer system. **1, 12-16**

software licence A legal agreement that states how software can be used and distributed. **16, 47**

solid state drive (SSD) Alternative to a traditional magnetic hard disk drive that uses flash memory. **8, 9**

sorting algorithm A set of instructions that you can follow to order a list of items. **57-59**
- **bubble sort 57**
- **insertion sort 59**
- **merge sort 58**

sound 112

source code The actual written code of a program. **16, 92**

spoof website A fake website that tricks users into thinking it's another well-known website. **32**

spyware A type of malware which secretly monitors and records user actions. **31**

SQL (Structured Query Language) A programming language used to manage and search databases. **32, 80**

SQL injection A piece of SQL code which can be typed into an input box to try and attack the SQL database. **32**

stakeholder Somebody who has an interest in or is affected by the decisions of an organisation. **38**

star topology A type of network topology where all devices are connected to a central switch or server which controls the network. **24**

static IP address A permanent IP address. **27**

storage device A device used to read and write data to a storage medium. **8, 9**

storage medium A thing that holds data. It can be part of the storage device (e.g. magnetic disks inside a hard drive) or separate (like a CD). **8, 9**

string A data type for text.
- **data type 63**
- **manipulation 67**

sub program A set of code within a program that can be called at any time from the main program. **81, 82**

surveillance The act of monitoring what people are accessing on the Internet. **39**

SWITCH-CASE statement A type of selection statement. **71**

switch (network) Connects devices together on a LAN and directs frames of data to the correct device. **21**

syntax error An error in the code where the rules or grammar of the programming language have been broken. **90**

system software Software designed to run or maintain a computer system. **12-15**

T

TCP/IP A set of protocols which dictate how data is sent over the Internet. Made up of Transmission Control Protocol (TCP) and Internet Protocol (IP). **27-30**

terabyte 1000 gigabytes. **100**

tertiary storage High-capacity external storage used mainly for back ups. **8**

test data Inputs that are chosen to see if a program is behaving as intended. **91**

testing A way of checking if a program functions correctly and meets certain requirements. **90, 91**

test plan A detailed plan of how a program is going to be tested including what test data will be used. **91**

topology (networks) How the devices in a network are connected together. **24**

traffic (networks) The amount of data travelling on a network. **28**

translator A program that turns a programming language into machine code. **92**

trojans A type of malware which is disguised as legitimate software. **31**

trolling The act of trying to provoke public arguments online. **40**

truth table A table listing all possible binary inputs through a logic circuit, with the corresponding outputs. **98, 99**

U

Unicode® A large character set that attempts to include all possible characters. **110**

units (of data) 100

URL (Uniform Resource Locator) An address used to access web servers and resources on them. **30**

user access levels Controls what files or areas of the network different groups of users can access. **33**

user account 14

user interface Provides a way for the user to interact with the computer. **13**

utility software Software designed to help maintain a computer system. **15**

V

validation Checking that an input meets certain criteria. **32, 87**

variable A named value which can be changed as the program is running. **66**
- **global 82**
- **local 82**

viral Content on the Internet which has spread rapidly via social media. **41**

virtual memory Area of secondary storage used by the OS as extra RAM. **6**

virtual network A software-based network that exists between devices on a physical network. **30**

virtual server A software-based server. **45**

virus A type of malware which spreads by attaching itself to files. **31**

volatile memory Memory that loses its contents when it has no power. **6**

Von Neumann A type of CPU architecture. **2, 3**

W

WAN (Wide Area Network) A network which connects networks in different geographical locations. **20, 30**

Waste Electric and Electronic Equipment (WEEE) directive 45

WHILE loop Type of iteration statement. **72**

Wi-Fi® The standard used for wireless connections between devices. **22**

WIMP A GUI based on windows, icons, menus and pointers. **13**

wired / wireless networks 20-22

wireless access point (WAP) A piece of hardware that allows devices to connect wirelessly. **22**

world wide web (www) The collection of websites hosted on the Internet. **30**

worms A type of malware which replicates itself. **31**

The definitive revision and practice books for GCSE

"We used to spend all our time worrying about exam results, but now we use CGP books and we know our students have the best revision materials available. Now our spare time is carefree."

Sara Short (left)
LEA Advisor (Numeracy)
Great Yarmouth

Isobel Monkhouse
LEA Advisor (Numeracy)
Great Yarmouth

www.cgpbooks.co.uk

0518 - 17471

CGP

OCR GCSE Computer Science is one tough cookie...

...don't worry — this all-in-one CGP book breaks it down into byte-sized chunks!

- **Everything you need to know for the exams...**
  It's all here, from flash drives to flow diagrams
- **Tech-speak decoded...**
  You bet — our notes are the clearest by miles
- **Exam-style questions for every topic...**
  And a full practice exam — amazing!
- **100% up to date for the Grade 9-1 course...**
  Well, obviously — it's a CGP book!
- **Complete with a free Online Edition...**
  Even more cache for your cash!

You don't need an algorithm to get your revision in order — CGP has it sorted ☺

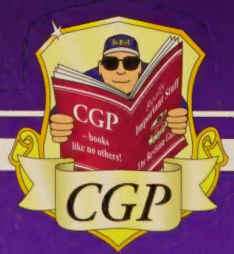
Not dishwasher safe

ISBN 978 1 78294 860 5

9 781782 948605

COS41

£10.99
(Retail Price)



www.cgpbooks.co.uk